DOCUMENT RESUME

ED 213 400 IR 010 022

AUTHOR Nordman, R.; Parker, J.

TITLE Teaching Computer Literacy in an Elementary School: A Comparison of Two Methods Using Microcomputers. Report No. 81:18.

INSTITUTION Educational Research Inst. of British Columbia, Vancouver.

PUB DATE Sep 81
NOTE 165p.; Best available copy.
AVAILABLE FROM Educational Research Institute of British Columbia, Suite 400, 515 West 10th Ave., Vancouver, B.C. V5Z 4A8 ($15.00).

EDRS PRICE MF01/PC07 Plus Postage.
DESCRIPTORS *Computer Literacy; Computer Programs; Educational Innovation; Elementary Education; Foreign Countries; Intermode Differences; *Microcomputers; *Programing; *Programing Languages; *Teaching Methods
IDENTIFIERS BASIC Programing Language; British Columbia

ABSTRACT

This report compares two methods of teaching BASIC programming used to develop computer literacy among children in grades three through seven in British Columbia. Phase one of the project was designed to instruct children in grades five to seven on the arithmetic operations of writing simple BASIC programs. Instructional methods included using job cards, program cards, and large group instruction, and each student was allowed to spend an equal amount of time on the computer. Phase two was also designed to teach arithmetic operations of BASIC programming, but this time worksheets and take home assignments were the major instructional tools used with children in grades three through five. Moreover, students were only allowed to use the microcomputers as needed to complete assignments. Results indicated that children from both groups made gains in their comprehension of programming, but that this was not necessarily related to age or instructional methods. It was concluded that LOGO type languages may be better for working with elementary school students, with BASIC being reserved for more advanced stages of instruction. This report includes pre- and posttests, scores, job cards for the lessons, and background material for teaching computer literacy prior to programming. (MER)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)
☒ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

ED213400

TEACHING COMPUTER LITERACY IN AN ELEMENTARY SCHOOL,

A comparison of two methods using microcomputers

This project was completed with
the financial assistance of a
research grant from the Educational
Research Institute of British Columbia

R. Nordman
J. Parker

River Bend School
Port Alberni, B.C.
Sept. 1981.

IR010022

BEST AVAILABLE COPY

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Reg Nordman
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

The Fall of 1979 brought the computer world to RiverBend Elementary, with the arrival of a TRS-80 model I Microcomputer. By the end of that school year, the students and staff had been exposed to computers through games, simulations and a whole school unit on computer literacy. The thrust for the 1980-81 school term was to increase the computer usage.

For the 80-81 term, John Parker joined the school, so that there were two programmers on staff. John had developed his interest in micros through electronics. He had purchased a TRS-80 and had been using it in a Grade three-four class. He had completed a local college course on BASIC programming.

Reg Nordman, already on staff, gained his programming experience as a mining engineer, working with an IBM System 3-mini computer.

Prior to applying for an ERIBC Grant, John and Reg decided to try to teach BASIC programming to a Grade 5,6,7 class. A lengthy periodical search failed to turn up research into effective use of computers in an elementary classroom. A few authors (Albrecht, Ahl) wrote how they did teach programming, but there was no number data saying how successful they were, nor a curriculum to follow.

Software was available, but the amount that would be actually usable in a classroom appeared very scanty. John and Reg also experienced problems in getting speedy, regular delivery of software. From their own library of software, the route to follow seemed to be ....

away from games, leaving programming as a viable alternative.

BASIC was readily available on micros. It had a simple vocabulary, and the language also produced results on the screen with a minimum of statements. The hardware on hand could not accommodate a more highly structured language without investment in discs, memory and software. As well, Reg and John were already very familiar with BASIC. (A note to the unfamiliar. A programme is a list of instructions put into the computer which it is to follow. A programmer makes up these lists. Hardware is jargon for TV screens, computer keyboards and anything else which hooks up to the computer. Software is books, magazines and programs recorded in the computer or on magnetic tapes or discs - anything the computer or you can read.)

Phase I of the project started in September, 1981. Three TRS-80's were in the class, one equipped with an Exatron Stringy Floppy. The unit was to run for eight weeks, as that seemed an amount of time which any teacher could find in a term. The students (grades 5-7) were given a startup demonstration on the micros and assigned to certain daily blocks of time on the micros. Job cards had been made up following Bob Albrecht's "My Computer Likes Me," articles in Creative Computing Age. (see appendix A1) The students typed in the prewritten programs. If they were careful typists, the program worked. Since their blocks of time were all equal, some students quickly outpaced the others in completing job cards. The teacher intervened when the student had problems with a program, usually a

typing error. Group instruction was to follow exposure to new vocabulary, however, the class was soon dispersed into individual groups. The content included PRINT, INPUT, RUN, LOAD, IF THEN, FOR NEXT, SET, RESET, GOSUB, RETURN, RND, PRINT AT.

Those students which finished the job cards were given programming problems to solve. (appendix A2) Some students, with the teacher's aid, were successful in building small, useful programs. The majority of the group appeared to have a low comprehension of what the words in BASIC did.

A vocabulary, plus programming test, was given to one half of the group. (appendix A3) The test was marked in class, demonstrated on the computer, and discussed by the whole group. A retest, also with programming questions, was set. Results showed an increase in comprehension which was very encouraging. (see appendix A4.) Whole group instruction with more testing seemed to increase word comprehension. However, there were still only a few true programmers in the group.

RESULTS - Phase I

Students were able to input more programs, more quickly by the end of the unit, than at the start. They could proofread and edit their programs without teacher aid. They could modify existing programs to fit their needs, by changing the parameters but not the structure. Students could analyse a simple problem and describe what the computer was supposed to do. Given a short programming assignment, few students could successfully complete it without teacher assistance. Student interest remained very

high during the unit.

Phase II - Phase II of the project started in January, 1981 with a new group of students (grades 3 to 5) This part of the study was aided by an Educational Research Institute of British Columbia grant. John and Reg were the Co-Directors of the project. Analysis of the Phase I results indicated a need for more whole group, interactive discussion to build comprehension of the language. The course content was to be cut down, but the eight week length was kept. Students would stay on the computer until they completed their task, which would hold the group into more of a unit. The teacher would use a micro, with three monitors, to initiate and review discussion of what a particular word, or words in BASIC did. Worksheets were built to be used with the computer and as short takehome assignments. By reading, typing in the program, and having a correct computer response, the student knew he was successful. Then, he was to change the program slightly to obtain a different computer response. The next assignment required the student to solve a programming problem, using what he had learned. If the program ran, he was successful. The student documented his programs before moving on. (see appendix B1.)

Short subjective pre tests on attitudes to computers were set. (appendix B2)

Tests on following directions, and story sequencing were set as pre and post tests.(appendix B3) The completed worksheets

were graded and recorded during the unit. During the whole group instruction, anecdotal notes were kept on each student's response. A test, similar to that set in Phase I was used as a culminating activity.(appendix B4) Each student was conferenced about why he chose certain answers on the final test. A directed essay was submitted by the students at the end of the unit (appendix B5)

The co-directors were looking to see skill improvements in programming, editing and proofreading, sequencing and following directions.

	PHASE I	PHASE II
TIME:	Eight weeks	Eight weeks
METHOD:	Job cards whole group instruct. program cards	worksheets, take home assignments, whole group, discuss, assign.
MACHINE USE:	Equal time; in order.	As student needed to complete worksheets.
DATA BASE:	Age 11-13 years Boys 10 Girls 15 Total 25 Reading level Gd. 4-11 Grade Level 5 to 7 0 micros at home 0 parents with U. degrees	9 - 11 years 6 13 19 3-9 3-5 0 micros at home 0 parents with U. degrees
EVALUATION:	Written test Essay on computers Programming problems.	Continuous with worksheets Pre/Post - attitude, sequencing Following directions, Post written test Programming test Essay questions Conferencing (individual)
CONTENT:	Arithmetic operations: Variables PRINT, INPUT, RUN, LOAD, IF THEN, FOR NEXT, SET, RESET, GOSUB, RETURN	Arithmetic operations, Variables PRINT, INPUT, RUN, LOAD, IF THEN FOR NEXT

COMPARISON OF PHASE I AND PHASE II

RESULTS - Phase II

Students developed accuracy and speed in typing in programs. They could analyze a multistep program more accurately, quickly, and confidently than the Phase I group. Interest was still very high. Pretesting showed fair skills in following directions. Some individuals were weak on sequencing, and attitudes to the computer were positive. (Appendix A4)

Early programming worksheet assignments were very successful (as opposed to Phase I results) which boded well for the method. As new words were added, programming became too difficult for many (60% were unsuccessful at writing a program requiring PRINT, INPUT, IF THEN and FOR NEXT.)

Final written test results indicated a low comprehension of BASIC words, however, individual conferencing showed that the students had a high comprehension of vocabulary and a good grasp of programming. The test had failed to measure that.

MEANING OF RESULTS

The results of the study are difficult to analyze. With the use of computers coexisting with all other facets of the curriculum, the Directors are unwilling to attribute many gains in

skills solely to computer use. The progress of the project could not be allowed to interfere with other classroom activities. A teacher has many jobs in a classroom and the project was designed so that programming could be incorporated into a daily schedule. A balance was to be maintained between computer and non computer lesson time. Due to this balancing, days would occur which did not allow for any computer time. Research has indicated continuity is important to good teaching technique. In a classroom, all subjects can expect breaks in their continuity. The idea was to make the classroom situation realistic. If anything, the computer work received more gaps in its continuity as the teaching of programming would point out other teachable areas of weakness in children. (A curriculum has to take into account teacher, student, and material variables to be given a successful usage.)

While exposed to the micros, student attitudes remained positive.

Gains in proofreading skills seemed subject specific without general transfer.

Large group interaction is more conducive to increasing the comprehension of program and the BASIC language parameters. This comprehension is an aid to programming, but actual programming requires a new level of thinking for the child.

Recent writing on computer languages bears out this idea of thinking. Seymour Papert feels this way about computer programming and thinking;

"By providing a very concrete, down to earth model of a particular style of thinking, work with the computer can make it easier to understand that there is such a thing as a "style of thinking', and giving children the opportunity to choose between one style or another provides an opportunity to develop the skill necessary to choose between styles. Thus instead of inducing mechanical thinking, contact with computers could be the best conceivable antidote to it. What is most important in this is that through these experiences these children would be..... learning to think articulately about "thinking"." His stance accepts the teaching of programming as a very useful technique. For the project just completed it does not answer why the learning of BASIC programming was so difficult for both groups of children."

The initial reaction was to analyze the tests. Did they test what was taught? Was there content validity? This first lookover showed that yes, some of the paper and pencil work defeated some children through format and style. This could explain the increase in comprehension shown in oral conferencing versus that demonstrated on a written test.

Similar questions were posed with respect to difficulty of

material, course structure, hardware limitations, room layout, teaching environment, group dynamics, and teaching environment, group dynamics, and teaching methods. These variables could be changed to perhaps remedy some of the difficulties the children experienced in learning how to program. (see appendix C1)

In reviewing difficulty of material the question of how difficult is BASIC was asked. With its simple vocabulary, it seemed easy to learn. The children very quickly were able to read a program, tell what it was to do and they could correct syntax errors. Is BASIC easy to use? How high a level of abstrac - tion are the children dealing with?

Seymour Papert argues that choosing the programming language is "consequential, as a programming language favors certain metaphors, images and ways of thinking." He believes BASIC is easy to learn. "But using it is a different matter. Programs in BASIC acquire so labyrinthine a structure that only the most motivated and brilliant (mathematical) children do learn to use it for more than trivial ends."

The RiverBend experiences support his statement, as the Grade 3 to 7 groups, with a few exceptions, found BASIC difficult to use, without a lot of teacher assistance. With the amount of preparation involved, class time, and student time on the micros, the Directors began to question the efficacy of teaching pro-

gramming to this age child. Any teacher cannot ignore Jean Piaget's work in this field, so the level of abstraction in BASIC could help defeat its instruction. Quite likely, some students in junior-senior high school would still be unable to "use BASIC effectively."

There are "simpler" languages, i.e. highly structured but easier to use, being developed. The Directors see a probable use for PASCAL-PILOT, LOGO, SMALLTALK, etc. as an elementary programming language, with BASIC being reserved for enriched or advanced students.

During the project, the Directors were continually struck by the similarities between teaching programming and any other subject. Once the teacher masters the language, its communication to students will utilize as much time, effort and art as any other curriculum. The technology in place supports the need for a continual computer literacy, but the same technology does not guarantee to lessen the work involved.

Many articles and books are beginning to appear on the uses of micro computers in the classroom. The Directors suggest that persons involved in course selection and materials do so with one foot planted firmly in the reality of the classroom, and the other keeping abreast of current computer language development.

Submitted to ERIBC

1981

APPENDIX INDEX

GROUP A - Grades 5-6-7

A1 Job cards used by students

A2 Examples of program cards to be used when the student is finished job cards.

A3 Test used to determine what students had learned about BASIC during 8 weeks.

A4 Test results.

A5 Methods used to sign up students for computer time.

A6 Basic Dictionary which could be used by students for extra programs.

GROUP B - Grades 3-4-5.

B1 Job sheets used by students.

B2 Pre attitudinal paragraph learning about computers.

B3 Pre test examples for following direction and sequencing.

B4 Test used to determine what students had learned about BASIC during 8 weeks test results.
Post test used for following directions and sequencing, test results.

B5 Post attitudinal paragraph questions
Post attitudinal paragraph results.

B6 Sample anecdotal comments on large group instruction.

B7 Sample of overhead transparency used in large group instruction (both groups)

EXTRA APPENDIXES

C 1 Write up about work shop given to local teachers about ERIBC grant. Hand out given at workshop

C 2 In district publications of project.

D Computer literacy booklet prepared 1979-80 by River Bend Elementary school which was sent to about 20 school districts.

E Annotated Bibliography of resources used to prepare for the computer literacy course 1980-81 including magazines, books and program tapes.

These Programming ideas were not used by many students because of time constraints. However, they could be used as problems to develop an understanding of the "BASIC" language, based on the computer dictionary.

1. Write a program that inputs a name then a birthdate. Then prints them.

2. Write a program that computes the area of a triangle.

4. Write a program which prints the odd numbers from one to twenty one.
5. Write a program which generates a random number between 1 and 12, then prints out whether the answer is even or odd.

5. Write a program which generates a random number between 1 and 12, then decides if it is even or odd. If the answer is even then the program prints the answer. If the number is odd the program goes back to get another random number.

5. Write a program which uses a GOSUB to print the numbers 1 to 10. Write an introduction that prints what your program does.

6. Write a program that inputs two numbers; then prints both numbers one on top of the other at the middle of the screen, then adds the two numbers together., then prints the answer below the two other numbers.
(you will need print @ statements)

6. Write a program that inputs two different words. Then prints the words one at each of two corners of the screen.

6. Write a program that prints a word at the top left hand corner of the screen, then waits 5 seconds. Clears the screen. Then prints the word at the top right hand corner of the screen. Then waits 5 seconds clears the screen. Then prints the word at the bottom right hand corner of the screen. Then waits 5 seconds. Then clears the screen. Then prints the word at the bottom left hand corner of the screen. Then starts again at the beginning of the program.
(for experts) change the above program so that it blinks faster and faster until it reaches maximum speed.

6. Write a program which prints your name five times one right below the other in the middle of the screen.

8. Write a program that generates three random numbers. Then decides which one is the highest, which one is in the middle and which one is the lowest. Then prints them out in the bottom right hand corner of the screen in descending order.

3
7
9

10. Write a program that draws a map for a game of Xs and Os. (use set and for next loops)

10. Write a program which draws a picture of a dog or a horse. (use set or print chr$)

10. Write a program which draws a picture of a house. (use set or print chr$)

10. Write a program that draws a picture of a car or an airplane. (use set or print chr$)

10. Write a program that draws a rectangle (hollow)

10. Write a program that draws a hollow triangle.

NOTE: Numbers indicate degree of difficulty - one - easy
ten - difficult.

Programming Ideas (continued)

7. Write a program that inputs four different words then prints them out in the form of a sentence.

7. Write a program that inputs four different words then outputs them in the form of four different sentences.

7. Write a program that generates a random number less than 99. Then inputs a new number. Then adds the two numbers. Then outputs the answer.

2. Write a program which inputs a number then multiplies it by itself (squares it) then outputs the answer.

2. Write a program which inputs a number then outputs the cube of the number.

2. Write a program that computes the area of a rectangle after information is inputed.

2. Write a program that will input the length, width and height of a box. Then print out the volume of the object.

3. Write a program which will give the perimeter of many sided object. (use a for next loop with an input statement to put in information.)

3. Write a program that inputs a name of a football team. Then the scores from each of the four quarters. Then prints out the average score per quarter.
(look up how to find the average)

4. Write a program that inputs two numbers then subtracts the smaller one from the bigger one then outputs the answer.(you may need an if then statement.)

4. Write a program which prints the even numbers from one to twenty. (you may need to use the step statement)

7. Write a program which inputs todays date. Then generates a random number between 1 and 6. Then multiplies the two numbers together. Then outputs an answer.

A3 What Do You Know about "BASIC" 1.

I The following programs or statements do what they are supposed to do. (are correct) circle T or F

1. 10 Print "man" T F
2. 7 Print "man" T F
3. 10 Print A$ T F
4. 10 A$= "man" T F
   20 Pring A
5. 10 Input A$
   20 Print B$ T F
6. 10 for I= 1 to 10
   20 Print "man"
   30 Next I T F
7. 10 Input B1$
   11 Print B$
   12 Goto 10 T F
8. 10 Print @ 1187, "man"; T F
9. 10 Goto 12
   12 Print "man" T F
10. 10 Print "man"
    15 Print "went"
    17 Print "home"
    The computer will print
    man went home T F
11. 10 cls
    20 A$ = "man"
    30 Print A$
    40 Goto 40
    The computer will print
    manmanmanman etc. for ever T F
12. 10 ForJ= 1 to 1000
    20 Next J
    This program is a loop that can be
    used to time an action. T F

13.
```
10 for I = 1 to 1000
20 next I
```
will last for a shorter time than
```
10 for J = 1 to 100: next J
```
T F

14.
```
10 cls
20 Input Z$
30 cls
40 Print Z$
50 Let Q = 100
60 For P = 1 to Q
70 Next P
80 cls
90 Let Y = 1000
100 For T = 1 to Y
101 Next T
130 60 goto 30
```
a) This program will run correctly T F

b) This program will blink the word you enter on and off till the break key is pressed. T F

c) The on time will be longer than the off time. T F

d) If you left out line 30 the program will still work properly T F

e) This program could be written with a GOSUB instead. T F

f) line 110 should read 110 next y T F

15.
```
10 cLs
20 Input A$
30 Print A$;
40 Input Y
50 For X - 1 to y
60 Next x
70 Goto 30
```
All statements are in the correct order to have this program work properly T F

Circle The Right Answer

16. SN error means
    a) The program has been incorrectly loaded
    b) The line does not have the correct punctuation
    c) You have not turned the computer on correctly
    d) You have typed the line in correctly.

17. 10 Print "man"
    20 Gosub 100
    30 End
    100 CLS
    110 Return
    a) will print man once
    b) will print man once then clear the screen then print ready
    c) will not work properly
    d) Does not need line 110
    e) Line 110 makes the program jump back to line 10

18. 5 A$ = "man"
    10 Print @ 84, A$;
    a) will not work.
    b) will print man on the second line.
    c) will print man on the first line.
    d) will print man at the center of the screen.
    e) will print Ø on the second line.

19. 5 B$ = "woman"
    10 N = 70
    20 Print @ N, A$
    a) will not print anything
    b) will print "man" on the second line
    c) will print Ø on the first line
    d) will print "man" on the first line
    e) will print Ø on the second line

Circle The Right Answer

20
```
10 cLs
20 Input "Type what you wish printed; C$
30 For K = Ø To  1Ø23
40 cLs
50 Print @ K, C$;
60 Next K
70 Goto 30
```
a) This program will print what you have typed at each of the 1023. Print positions so that the whole screen is filled.

b) This program will print what you have typed at each of the 1023 print positions after clearing each position.

c) This program will not work properly

d) If you change line 70 to Goto 10 it will work properly (better.)

e) K is the string variable that tells the computer where to print each time.

21.
```
10 cLs
20 Set (130,58)
```
a) will print a square at the bottom right corner.

b) will print a square at the top left corner

c) will print a square at the top right corner.

d) will print a square at the bottom left corner.

e) is not a correct program

22.
```
10 x = 10
20 If x>10 then 40
30 If x > 9 then 50
40 Print "man";
50 Print "woman";
```
a) the program prints man

b) the program prints nothing then says ready

c) the program won't run.

d) the program prints woman.

Find and fix the two errors in this program

23.
```
10 ¢Ls
20 Input "what is your name"
30 ¢Ls
40 X=Rnd (1023)
50 Print @ X, N$
60 T=100
70 For Z= 1To T
80 Goto 30
```

24. Write a program that will print your name near the center of the screen.

25. Write a program which will print a square somewhere on the screen then clear the screen then print the square again after a period of 100 units.

26. Write a statement that tells how you feel about what you have done with the computer this term. (on the back of this sheet)

27. Write a statement that tells how you could have learned more about how the computer works. (on the back of this sheet)

TEST METHOD 1 RESULTS

STUDENT:	TEST 1	TEST 2	Δ
A	10	15	+5
B	16	16	+0
C	9	21	+12
D	10	22	+12
E	8	24	+16
F	8	16	+8
G	6	13	+7
H	12	21	+9
I	15	15	+0

WHOLE GROUP	TEST 2	TEST 1
	No. of students	(1/2 group) No. of students
MK 25	3	
20-24	5	
15-19	7	1
10-14	3	5
0-9	0	6

Group 1

Appendix A4-1

Attitude question (post test results)

How would you learn more?

- more time in class 2
- more help on the computer 2
- more reading and thinking about the words 8
- learned a lot 16
- feel good 11

Was it fun?

- Yes 5
- No 1

comments

Want one of my own.
It makes you think.
I want to experiment with the computer.
I want to take it apart now.
I should have listened to the lessons more.

Sign up for computer time.

During the year, we used a number of different methods to sign students up for computer time.

For Group A we used a signed up sheet which the teacher made from the class list. Each student got roughly the same amount of time on the computer.

For Group B we used teacher judgement to sign up students who were not yet finished the assignment. This meant that some students got more computer time than others.

For lunch and after school sign up times we used three different methods.

1.) The students signed themselves up on the regular sign up sheet in the blank lunch and after school spaces. There was some trouble with students erasing names.

2.) The teachers were asked for computer time which he noted in a book. Occasionally the teachers made a mistake and signed up two students for one computer.

3.) The students were asked to daily place masking tape with their name and time on it on the computer. This method seemed to work best.

COMPUTER DICTIONARY "BASIC"

RUN

Run will Run program

Run 70

will start at line 70 then run program. Note it may not work properly

```
10 Input A
20 Input B
30 Print A+B
40 Run
```

This program will start over again when it gets to line 40

NEW

gets rid of all of your program so you can't run or list.

(caution) Make sure you don't need your program. Check with your teacher to make sure that you are finished.

ARITHMETIC OPERATORS

\+ addition
\- subtraction
* multiplication
/ division
↑ exponentiation $2\uparrow 2 = 2^2 = 4$

$2\uparrow 3 = 2^3 = 8$

< less than
> greater than
= equal to
<> not equal to
<= less than or equal to
>= greater than or equal to

VARIABLE NAMES

A to Z (used for numbers)
(AØ - A9) - ZØ - Z9)
(A A, A B, A C...A Z) - ZA - ZZ)

(For whole number (integer) variables include % after A%, A9%, ZQ%)
Special Purposes)

STRING VARIABLES (used for words or expressions)

A$ ⟶ ZZ$

PRINT

10.
```
10 Print "mark";
20 Print "went home."
```
(NOTICE SEMI-COLON MAKES NEXT LINE CONTINUE ON. HIT SPACEBAR AFTER THE "K" IN MARK THEN PUT IN QUOTATION MARKS TO MAKE IT LOOK RIGHT.)

11.
```
10 Print "g",
20 Print "h",
30 Print "o",
40 Print "s",
50 Print "t",
```
(NOTICE THAT THE COMMA AT THE END PRINTS A LETTER IN EACH OF THE FOUR ZONES (ALSO CALLED FIELDS) THEN STARTS AGAIN LOOK AT PROGRAM 4)

12. Make the following programs correct by adding the necessary parts:

a) 10 Print "the dog has a bone"
b) 10 Print the cat ran home
c) 10 "January"

12. (continued)

d) "1 ø Print candy is bad for your teeth."

e) Print "Help"

f) Print 5 apples are left

g) 10 Print 10+5=

h) 10 Print "11, "John"

13. Write a program which will give your name, grade, school, and teacher one in each zone or field. LOOKAT PROGRAM 4, 9, 11.

14. Write a three line story using at least one variable. Use line numbers 10,20,30,40. LOOK AT PROGRAMS 5,7,8.

PRINT

Uses: labels, names, headings, statements.

1. 10 Print "John"

2. 10 Print "RiverBend Basketball scores"

3. 10 Print "2+7=9" NOTICE

4. 10 Print "John","Mary", "George", "Henry", "Patty"

**** (NOTICE THE COMMA PUTS EACH NAME IN ONE OF THE FOUR ZONES ACROSS THE SCREEN THEN STARTS OVER AGAIN AT ZONE 1)

5. 10 A=5
20 Print "there are"; A;" boys here"

**** (NOTICE THE 2 SETS OF QUOTATION MARKS SEPARATED BY THE SEMI-COLONS AROUND THE VARIABLE A.)

6. 10 A=5
20 B=6
30 Print A+B

**** (NOTICE THE COMPUTER ADDS THE VARIABLES A AND B TOGETHER AND PRINTS THE ANSWER.)

7. 10 A=5
20 B=6
30 Print "there are "; A*B;"Girls at camp"
(LOOK AT PROGRAM 6 NOTICE THE SIMILARITIES

PRINT (continued)

8.
```
10 A=5
20 B=6
30 Print A;"people are in each row.  There are "'B'rows"
```

9.
```
10 Print "John"," ", "  ", "Mary"
```
*** LOOK AT PROGRAM 4 WORDS INSIDE QUOTATION MARKS MAYBE LEFT OUT TO SKIP ZONES.

INPUT

1.
```
10 Input A (INPUT FIRST NUMBER )
20 Input B (INPUT SECOND NUMBER)
30 Print "sum of ";A;" and ";B;" is ";A+B
```
A AND B ARE ADDED AND PRINTED

2.
```
10 Input A,B (INPUT FIRST, THEN SECOND NUMBER)
20 Print "sum of "; A;" and ";B;" is ; A+B
```
A AND B ARE ADDED AND PRINTED: NOTICE THIS PROGRAM DOES THE SAME THING AS PROGRAM 1.

3.
```
10 Print "How many funfair raffle tickets were there at the
start of the week,"
20 Input I PUT IN STARTING NUMBER OF TICKETS
30 Print "what were the sales for each day of the week?"
40 Input M,T,W,Th,F.(INPUT FIVE NUMBERS)
50 Print (SPACES PROGRAM 1 LINE DOWN
60 Print "Beginning inventory";I; "tickets"
   (INVENTORY MEANS WHAT YOU HAVE IN STOCK)
70 Print (SPACES PROGRAM 1 LINE DOWN)
80 Print (SPACES 20 SPACES) Daily sales"
90 Print (SPACES 20 SPACES) - - - -MAKES PROGRAM NEATER
100 Print "Mon.", Tues.", "Wed.", "Thurs.", "Fri."
110 Print "M,T,W,Th,F" (NOTICE FRIDAY WILL BE ONE LINE LOWER)
120 Print "number of tickets sold this week" ;M+T+W+Th+F
   ADDS ALL TICKETS SOLD DURING WEEK
130 Print "ending inventory"; I-(M+T+W+Th+F)
   SUBTRACTS TICKETS SOLD DURING WEEK FROM NUMBER
   AT BEGINNING OF WEEK.
140 End
```

INPUT

```
10  Input "What is your name?"; A$
20 Print A$;" I like you"
```

YOU MAY INPUT A WORD OR STRING VARIABLE

4.
```
10 Input "what is your name and age?;" A$,A
20 Print A$,A
```

STRING VARIABLES AND VARIABLES CAN BE MIXED IN AN INPUT STATEMENT.

5.
```
10 Input A
20 Input B
30 Print A+B: Print A-B: Print A/B: Print A*B
```

SEE PROGRAM 1 THE COLON":" ALLOWS YOU TO PUT MORE THAN ONE COMMAND ON A LINE.

IF STATEMENT: (the "tester")

This is a decision statement. If a condition is true another operation is done. If the condition is false the operation is not done.

ENTER and RUN

```
1ØØ CLS
11Ø A=5
12Ø IF A=5 THEN150
13Ø PRINT "A DOES NOT EQUAL 5"
14Ø END
15Ø PRINT "A EQUALS 5"
```

THE SCREEN SHOULD SHOW A EQUALS 5

Line 11Ø tells the computer A=5

Line 12Ø is an IF THEN statement which sends the computer to line 150 only if A=5, skipping over line 13Ø; 14Ø. We don't use them in this case.

Change line 11Ø to

11Ø A=6 and RUN

This should say

A DOES NOT EQUAL 5.

Line 11Ø said A=6
Line 12Ø tests A. If A=5, the computer zips to Line 150/.
But, the test failed, A does not equal 5, so the computer goes on normally, next line 13Ø and 14Ø. Why must we include line 14Ø? Because if not, the computer would continue on to line 15Ø, and we don't want that.

The IF THEN statement can use:

= equals
> greater than
< less than
<> is not equal to
<= is less than or equal to
>= is greater than or equal to.

GUESSING GAME!

```
10Ø CLS
11Ø Print "I WILL THINK OF A NUMBER.  MY NUMBER WILL BE 1 OR 2"
12Ø Print "GUESS MY NUMBER !!!!!!!!"
200    X=RND(2)       (NOTE:  THE COMPUTER CHOOSES A RANDOM NUMBER.)
300 Print: Input "YOUR GUESS (1 or 2)";G
31Ø If G   THEN PRINT "FOOLED YOU! THAT'S NOT MY NUMBER."
320 If G =  THEN PRINT" THAT'S IT!! YOU MUST HAVE ESP!"
4ØØ T = 2ØØ
41Ø For Z-1 to T: Next Z
5ØØ Go to 1ØØ
```

GOTO

The program jumps to the line mentioned

```
10  A=7
20  B=8
30  C=A+B
40 Goto 60 (GOES TO 60)
50 End (THIS PROGRAM NEVER GETS TO LINE 50)
60 Print C (PRINTS ANSWER)
70 Goto 10 (GOES TO 10 TO START PROGRAM ALL OVER AGAIN
            YOU MUST PRESS BREAK KEY TO STOP)
```

END

Stops program from going past the end line.

```
 10 Gosub 100
 20 A=7
 30 B=8
 35 Gosub 100 (GOES TO LINE 100 FOR INFORMATION)
 40 Print C   PRINTS C (15)
 50 End       STOPS PROGRAM
100 C=A + B
110 Return    GOES BACK TO LINE 40
```

FOR .....NEXT

1. USES - TO COUNT TIME (DELAY)

```
10 Print "Hi__"; (PRESS SPACE BAR OR THE WORDS WILL RUN
                  TOGETHER.)
20 For I=1 to 150 (NOTE HOW LONG IT TAKES FOR EACH WORD
                   TO APPEAR)
30 Print "There_";
40 For I = 1 to 300: Next I (NOTICE COLON FOR MORE THAN ONE
                             IDEA OR STATEMENT ON A LINE)
50 Print "How"_; (NOTICE SEMI COLON MAKES NEXT WORD CONTINUE
                  ON THIS SAME LINE.)
60 For I=1 to 600:  Next I
70 Print "are_";
80 For I = 1 to 1200:   Next I
90 Print "you"
```

TRY SOME OTHER NUMBERS IN LINE 80 TO SEE HOW LONG THEY LAST. NOTICE YOU MUST USE A VARIABLE (I). I AND J ARE OFTEN USED BY MANY PROGRAMMERS. NOTICE EACH FOR MUST HAVE A NEXT

2. TO LOOP OR COUNT THROUGH THE SAME STEP MORE THAN ONE TIME.

2. 10 For I = 1 to 10
20 Print "I Win" (NOTICE WITHOUT SEMI COLON AT THE END YOU GET 10 "I win" UNDER EACH OTHER)

3. 10 For I = 1 to 7
20 Print " 1 win_"; (Space NOTICE WITH SEMI COLON YOU GET 7 " I win's IN A ROW)

4. 5 Input "Input a number between 1 and 12"; A
10 For I = I to 12
20 Print I; "Times"; A;" equals"; I*A

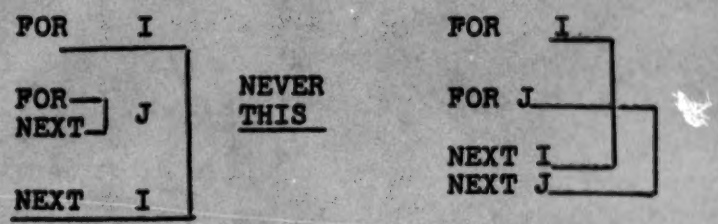
(NOTICE THIS LINE WAS FORGOTTEN AND PUT IN LATER. THIS IS THE REASON FOR NUMBERING LINES 10 APART.)
(NOTE THERE IS SOMETHING WRONG WITH THIS PROGRAM AS IT NOW STANDS.-CHECK THROUGH PROGRAMS 1-3 TO SEE WHAT IT IS BEFORE YOU LOOK BELOW)

40 Next I
NOTE LINE NUMBER COULD HAVE BEEN 30 OR IN FACT 10000.

3. FOR NEXT LOOPS CAN BE CONNECTED TOGETHER IN THE FOLLOWING MANNER.

5. 10 For I = 1 to 12
20 For J = 1 to 12
30 Print 1*J; (MULTIPLIES VARIABLES I AND J TOGETHER
40 Next J NOTICE LOOPS MUST HAVE THIS STRUCTURE
50 Next I NOTICE THIS PROGRAM GIVES YOUR TABLES (SORT OF JUMBLED)

(PUT IN THE 45 PRINT - NOTICE HOW THIS CHANGES YOUR PROGRAM. IS IT BETTER?)



For three loops
For I
For J
For K
Next K
Next J Next 1

6. Change Program 5

STEP COUNTS BY THE NUMBER OF STEPS STATED.

Change line 10 For I = 1 to 12 step 4

WHAT HAPPENS?

Change line 20 For J = 1 to 12 step 4

WHAT HAPPENS?

TRY OTHER STEP SIZES 1,2,3,4,6,8,12, OR THOSE IN BETWEEN

TRY line 10 For I = 12 to 1 step - 1

TRY OTHER NUMBERS

READ .....DATA

10 Remember this program prints the average
(THIS LINE IS NOT NECESSARY TO RUN PROGRAM)

20 Read X
(THIS LINE READS DATA FROM LINE 80 ONE PIECE AT A TIME)

30 If X = 9999 Then 70
(STOPS LOOP STARTED IN LINE 60 THEN PRINTS OUT ANSWER AT LINE 70)

40 S=S+X
(S IS THE SUM OF THE ADDITION OF ALL DATA STATEMENTS READ SO FAR. THE FIRST TIME S=0 BUT IS CHANGED BY S=0+5 S NOW EQUALS 5. IN THE SECOND LOOP S = 5+7 S=12 ETC.)

50 C=C+1
COUNTS THE NUMBER OF TIMES THAT DATA HAS BEEN READ WHEN S=5. C=0+1→C=1 WHEN S=12 C=1+1→ C=2)

60 Goto 20

(THIS LINE MAKES A LOOP TO READ EACH PIECE OF DATA ONE AT A TIME.)

70 Print S/C " is the average of ";C;" numbers ";
(SUM IS DIVIDED BY THE NUMBER OF DATA STATEMENTS READ (7)
9999 DOES NOT COUNT AS LINE 30 GOES TO 70 NOT TO 50)

80 Data 5,7,3,9,27,54,31,9999
(DATA LINE USUALLY COMES AT THE END OF A PROGRAM SO THAT IT CAN EASILY BE CHANGED. WRITE A NEW DATA LINE FOR THIS PROGRAM WHICH WILL PRINT OUT THE AVERAGE OF FIVE NUMBERS - REMEMBER TO PUT IN COMMAS BETWEEN EACH DATA STATEMENT)

90 End

GOSUB .....RETURN

IS USED WHEN YOU WANT TO DO THE SAME THING OR GROUP OF THING MORE THAN ONE TIME.

7 GOSUB 30 MEANS GO TO LINE 30 THEN DO THE THINGS ON LINE 30 AND FOLLOWING LINES UNTIL THE COMPUTER MEETS THE WORD RETURN. THE COMPUTER THEN GOES BACK TO THE NEXT LINE AFTER THE LINE WITH THE GO SUB COMMAND.

```
1    Input "Do you wish to (1) add (2) subtract
     (3) multiply or (4) divide"; A
2    If A=1 then 7        SENDS TO ADD
3    If A=2 then 9        SENDS TO SUBTRACT
4    If A=3 then 11       SENDS TO MULTIPLY
5    If A=4 then 13       SENDS TO DIVIDE
6    Goto 1               SENDS BACK TO START IF NUMBERS
                          OR KEYS OTHER THAN 1-4 ARE PRESSED.
7    Gosub 30             GOES TO 30 TO INPUT B THEN C
```

```
 8      Print B+C: Goto 1        GOES BACK TO START FOR NEXT
                                 QUESTION.
 9      Gosub 30
 10     Print B-C Goto 1         PRINTS ANSWER FOR SUBTRACTION
 11     Goxub 30
 12     Print B*C: Goto 1
 13     Gosub 30
 14     Print B/C: Goto 1
 30 Input B                      INPUTS FIRST NUMBER THE BIGGER
 31 Input C                      ONE INPUTS SECOND NUMBER
 32 Return                       (SMALLER) GOES TO 8,10,12, or
                                 14 ONE LINE DOWAN FROM 7 9 11 or
10 Input "Are you a student? (Y/N)"; A$ (NOTICE STRING VARIABL
20 If A$="N" then PRINT "Then you must be out of school"
                    NOTICE QUOTATION MARKS
30 If A$=Y" Gosub 60    GOES TO LINE 60
40 Print "keep learning"
50 End
60 Print "Try to answer this question"
70 A=Rnd (12): B=Rnd (12)
   (MAKES A RANDOM NUMBER BETWEEN 1 AND 12)
80 Print "What is"; A;"times";B
                         NOTICE SEMI COLONS
90 Gosub 180             GOES TO LINE 180 TO GET YOUR ANSWER
```

```
100   If C=A*B Print "Right"      (TELLS YOU IF YOU ARE RIGH
110   If C <> A*B Then Gosub 140  (GOES TO 140 TO GIVE YOU
                                   AN EASIER QUESTION)
130 return      (GOES TO 40 TO STOP PROGRAM IF YOU GET ONE
                 RIGHT )
140   Print "What is";A;"plus"; B
150 Gosub 180       INPUTS YOUR ANSWER
160  If C=A+B then 200     GOES TO 200 TO TELL YOU YOUR
                           RESULT.
170  If C <>A+B Print "Let's try a different one
175   Goto 30            STARTS YOU ALL OVER AGAIN  YOU MUST
                         GET ONE RIGHT TO STOP PROGRAM
180  Input C             GIVE YOUR ANSWER
190  Return              GOES BACK TO LINE 100 OR 160
200  Print "Good";Return     GOES BACK TO 40
```

ON.....GOTO

```
 2 Gosub 90          GOTO 90 GET A RANDOM NUMBER (1,2 or 3)
 5 Input A,B         YOU PUT IN 2 numbers. A AND B
10 On x goto 30, 50,70
                    (IF X=1 Goto 30 IF X=2 GOTO 50
                     IF X=3 GOTO 70 )
20 C=A*B

30 Print C.      ( IF THE RANDOM NUMBER X EQUALED 1 PRINT C)
40 End           STOP PROGRAM
50 Print A       (IF THE RANDOM NUMBER X EQUALED 2 PRINT A)
60 End
70  Print B      (IF THE RANDOM NUMBER X EQUALED 3 PRINT B)
```

ARITHMETIC OPERATORS

The computer uses the following arithmetic operators. Read the left hand side of the paper first. Then read the right hand side of the paper for any explanations. Please refer to lesson pages when answering all questions. Remember to keep notes on these pages of any extra information you are given in class.....

I A

ARITHMETIC OPERATORS

+ addition	
- subtraction	
* (star) Multiplication	note
/ (slash) Division	note
↑ (up arrow) exponentiation or powers	2 ↑ 2 = 4 2 ↑ 3 = 8
< less than	
> greater than	
= equal or equivalent	
<> not equal to	
<= less than or equal to	
>= greater than or equal to	

VARIABLES NAMES

A to Z	
(AØ-A9) to (ZØ-Z9)	Used for numbers only
(AA-AB-AC to AZ) to (ZA-ZZ)	A1 = 3 BZ = 4 A1 +B2=7

ARITHMETIC OPERATORS .....continued

STRING VARIABLES

A$ to Z$ AA$ to ZZ$

NOTE: Quotation marks must be used.

NOTE: the $ sign used for letters or words or sentences.

B$="A" CD$=" the man"

WORK SHEET

ARITHMETIC OPERATORS AND VARIABLES

1. What are the answers to the following arithmetic questions?

   a) 7+5= b) 8-4= c) 3*4=

   d) 14/2= e) 3*5-2= f) 16/4+2=

   g) 144/12 > 4*5 Circle T or F

2. What are the answers to the following questions?

   a) B=4: C=5 what is B*C =

   b) A1=7: Z7=6 what is A1 +Z7=

   c) AQ=10: D=4 what is AQ+D=

   d) A=56: Q4=8 what is A/Q4=

   e) P=64: Z2=8 what is C if C= P/8

   f) B1=55:C=35: S=14 what is P if P=B1+C+S

   g) What is wrong with this statement:

      A=5: B=25: C=50
      A*B = C

3. a) A = man — this is a legal statement T or F

b) P$ = play — this is a legal statement T or F

c) A = 7+4 — this is a legal statement T or F

d) B$ = "A" — this is a legal statement T or F

e) C$ ="3-1" — this is a legal statement T or F

f) B$ ="the man came home." — this is a legal statement T or F

PRINT

The word Print tells the computer to put information on the screen. You must give the computer information so that it can put it on the screen or the computer will print a blank line.

Try the following programs on the computer. Remember when you use line numbers you must type run to make your program start. Remember to press Enter at the end of each program line (when there is another line number it is the next program line)

1) 10 Print "John"

2) 10 Print "RIVER BEND SCORES"

3) 10 Print "2+7=9"

4) 10 Print "John,Mary,George,Henry"

5) 10 Print "John"; "Mary"; "George"; "Henry"
(using; spaces names)

6) 10 Print "John","Mary","George","Henry"

Program

1 NOTE: THE PRO-
2 GRAM PRINTS WHAT
3 IS IN QUOTATION
4 MARKS

5 NOTE: HOW THE COM
PUTER PRINTS. COM
PARE THIS TO PRO-
GRAM NO.4

NOTE: HOW IT PRIN
PROGRAM 6. COMPAR
THIS TO PROGRAM N
4 AND 5

NOTE: USING, PUTS
EACH NAME IN ONE
FOUR ZONES OR
FIELDS

ARITHMETIC OPERATORS WORK SHEET.....continued

7)
```
10 A=5
20 Print "there are"; A;"boys here."
```
NOTE: LOCATION OF QUOTATION MARKS.

NOTE: THE USE OF SEMI-COLON; AROUND VARIABLE

8)
```
10  A=5
20  B=6
30 Print  A+B
40 Print  A*B
50 Print A+B+A
```
NOTE: COMPUTER ADDS VARIABLES A AND B TOGETHER AND PRINTS ANSWER IN LI 30. THEN MULTIPLIES AND PRINTS A AND B TOGETHER IN LINE 40. THEN ADDS AN PRINTS A + B + C TOGETHER IN LINE 50

9)
```
10  A=7
11  B=4
12  Print "there are"; A*B;"girls
    at camp."
```
NOTE: ARITHMETIC OPERATIONS CAN BE DONE INSIDE SEMI COLONS IN A PRINT STATEMENT

10)
```
10  A = 4
40  B=9
70  Print there are"; A;" people in
    each"
90  Print "row. There are";B;"rows"
```
NOTE: VARIABLES ARE REPLACED BY THEIR VALUE WHEN THIS PROGRAM IS RUN

11)
```
10 Print "John","","","Mary"
```
No space bar

12)
```
10 Print "mark";
```
Press space bar

NOTE: PROGRAM 6 WHAT IS SIMILARITY? NOTE: SEMI COLON MAKES NEXT LINE CONTINUE. TRY WITHOUT SPACE BAR. WHAT HAPPENS?

ARITHMETIC OPERATORS WORK SHEET.....continued

13) 10 Print "G";
20 Print "H";
30 Print "O";
40 Print "S";
50 Print "T";

NOTE: TRY THIS PROGRAM USING A COMMA INSTEAD OF A SEMI COL TRY WITHOUT USING EITHER.

14) Make up your own program using print statements, variables and arithmetic operators

NOTE: HAVE A LOOK AT PROGRAMS 8,9 and 10.

"PRINT" WORKSHEET QUESTIONS

1) Make the following programs run correctly by adding the right information or punctuation. Do any of them run correct as typed?

a) 10 Print "The dog has a bone"
b) 10 "Print the cat ran home"
c) 10 "January"
d) "10 Print Candy is bad for your teeth"
e) Print "Help"
f) Print 5 apples are left
g) 10 Print 10 + 5 =
h) 10 Print 10 Print

2) a) Write a program that will print (bananas) (carrots) (apples) (nuts) one word in each of the four fields (see programs 4,5, or 6)

b) Write a program that will place two numbers as variables in the computers memory then print out one number divided by the other (see programs 8,9 or

*c) Write a program that prints a short story.use a string variable for each sentence you use. (this will require a great deal of thinking. Check the bottom of Page 1 for a clue also check programs 10 and 12.)

INPUT

The computer uses the word Input to take in information from the keyboard. When you run your program you will see a question mark for each Input statement. You must put in information, either numbers or letters or words or sentences, whatever is asked for by the variable, into the computer before it will do anything. After you have typed your answer you must press enter so that the computer will take in your answer and use it in the program. Remember to type run after entering your program.

1)
```
10 Input X
20 Input Y
30 Print X*Y
```

NOTE: ONLY A ? MARK SHOWS ON THE SCREEN THIS IS NOT USEFUL UNLESS YOU KNOW WHAT THE COMPUTER IS ASKING FOR.

NOTE: LINE 30TYPE UNTIL THE LAST Y BEFORE PRESSING ENTER

INPUT.....continued

2)
```
10 Input "type a number"; X
20 Input "type another number"; Y
30 Print "Your numbers multiplied
together give the answer";X*Y
```

NOTE: THE USE OF VARIABL FOR NUMBERS
NOTE: THIS PROGRAM DOES THE SAME THING BUT MAKES IT EASIER FOR THE USER TO UNDERSTAND WHAT IS WANTED. NOTE: PUNCTUATIO

3)
```
10 Input "what is your name";A$
20 Print "my name is computer"
30 Print "your name is";A$
```

NOTE: THE COMPUTER NOW SOUNDS INTELLIGENT

4)
```
10 Input "what is your name";A$
20 Input "what is your school";B$
30 Input"How old are you"; C
40 Print"Hi";A$;" It is nice to
meet an ";C;" year old who goes to
"; B$;" school."
```

NOTE THE USE OF A STRING VARIABLE FOR A WORD

NOTE: ANSWER.
NOTE LINE 10 AND 20 USE A STRING VARIABLE
NOTE LINE 30 USES A VARI BLE, SO YOU MUST ENTER A NUMBER

INPUT

5)
```
10 Input "Input two numbers"; A,B
20 Print "the sum of ";A;"and";B;"
is"; A+B
```

NOTE: YOU MUST PRESS ENT AFTER EACH NUMBERYOU ENT OR YOU MAY ENTER ONE NUM BER, PRESS COMMA THEN TH NEXT NUMBER. EXAMPLE ? 104, 5
NOTE YOU MUST TYPE THE ANSWER IN THE CORRECT OR OR YOU WILL GET AN ERROR MESSAGE.

6)
```
10 Input "type your name. Type
your age"; A$,A
20 Print "your name is";A$
30 Print "you are";A;" years old."
```

NOTE: INPUTS TWO NUMBERS WITH TWO VARIABLES AND TWO ? MARKS

INPUT - TEST

1. Find the errors in the following programs and fix them.

a) 10 Input "What is your name"; A

b) 10 Input "type a number"; A$

c) 10 Input "type two numbers"; A$,B

d) 10 Input "type in your name, then your age"; A,B$

e) 10 Input How old are you

f) 10 Input

g) 10 Print "What is your name?";
20 Input A

h) 10 Input "How many eggs are there in a dozen"
20 Print X

i) 10 Input "How many cars are parked outside?"A$
20 Print "there are"; A; "cars outside."

2. a) Write a program that Inputs two numbers then prints them on the screen.

b) Write a program that Inputs a word and a number, then prints them on the screen in the first two fields.

c) Write a program, that asks forthe name of the day and the date, then prints out the date and the day of the week.

IF.....then

The words "if.....then "allow the computer to make a decision about information. If a condition is true then the computer must do something or go to a special line number.

1.
```
10 Input "type yes or no"; A$
20 If A$ = "yes" then Print "yes"
30 If A$ = "No" then Print "No"
40 Print "you didn't type yes or no"
```

NOTE: FIRST THERE IS AN IF WHICH DECIDES IF STRING VARIABLES A$ = THE CORRECT WORD F(LINE 20 ("YES") NOTE: NEXT THERE IS A THEN WHICH TELLS T COMPUTER WHAT TO DO NEXT. IN THIS CASE PRINT "YES"

2.
```
10 Input "type the number 1 or 2";
20 If A=1 then Print "you typed one"
30 If A=2 then Print "you typed two"
40 Goto 10
```

NOTE: GOTO MAKES THE COMPUTER GO BACK TO LINE 10 EACH TIME PRESS BREAK KEY TO STOP

3.
```
10  A=5
20  If A=5 then 50
30  Print "A does not equal 5"
40  End
50  Print "A = 5"
```

NOTE: END TELLS THE COMPUTER TO STOP RUNNING THE PROGRAM NOTE: LINE 20 IS TRUE SO THE COMPUTER SKIPS LINES 30 AND 40

4.
```
10 Input "type a number between 1 and 10";A
20 If A>5 then 50
30 If A<5 then 70
40  End
50 Print "the number you chose was greater than five"
60 END
70 Print "The number you chose was less than five"
```

CHANGE LINE 10 to A=6
WHAT HAPPENS?

NOTE: DECISION USES THE GREATER THAN AND LESS THAN ARITHMETIC OPERATORS. SEE SHEET 1.

IF.....THEN ...continued

Note: You have learned the word "GOTO" which makes the computer go to the line number after the word GOTO (see program 2)

Note: you have learned the word "END" which tells the computer to stop.

5.
```
10 Print "Happy";
20 End
30 Print "Birthday"
```

NOTE: PROGRAM WILL NOT PRINT BIRTHDAY. IT STOPPED AT LINE 20.

6)
```
10 Input "type Hi or bye";Z$
20 If Z$ = "Hi" then 50
30 If Z$ = "Bye" then  rint "Bye"
40 End
50 Print "Hi there friend"
60 Goto 50
```

NOTE: PRESS BREAK KEY TO STOP HI THERE FRIEND. NOTE BYE STOP PROGRAM AT LINE 40

NOTE RUN PROGRAM TWICE - ONCE TYPE HI, ONCE TYPE BYE.

NOTE: THE COMPUTER NEVER GETS TO LINES 50 OR 60, CHANGE LINE 40 TO 40 PRINT. WHAT HAPPENS WHEN YOU RUN THE PROGRAM TO LINE 50. WHAT HAPPENS WHEN YOU INPUT HELLO INSTEAD OF HI OR BYE?

7)
```
10 Input "type a number between 1 and 10";B
20 If B>7Print "sorry I don't know my eight,
nine, or ten times tables"
30 Input "type a number between 1 and 7";
40 Print C;" times"; B;"equals";C*B
```

8) Write a program that inputs a number between 1 and 3 then tells what number you typed in.

NOTE: USE IF....THEN SEE PROGRAM 2

9) Write a program that decides between three words that you typed then tells what word you typed in.

NOTE: USE IF....THEN SEE PROGRAM 6.

IF THEN TEXT EXERCISES

1) Fix the following programs.

a) (3 errors)

```
10 Input "type a number between 1 and 4
20 If B> 2 Print "your number is 3 or 4"
30 If B< 3 Print "your number is 2 or 1"
```

b) (2 errors)

```
10 A$ = 7
20 If A$ = 7 then 40
30 End
40 Print "A equals 7"
```

c) (3 errors

```
10 Input "type yes or no"; A$
20 If A$ = "yes" then 60
30 If A$ = "no" then 90
40 End
60 Print "you typed no"
70 End
80 Print "you typed yes"
```

d)

```
10 Print "you"
20 Goto 10
30 Print "sweet"
40 End
50 Print "kid"
```

MAKE THIS PROGRAM SAY
YOU
SWEET
KID
WHEN IT ISRUN.

2. a) Write a new program that uses two If....Then statements to decide about something that has been typed into the computer - use at least one End statement. See Program 1,2,4, or 6.

b) Write a program with at least one If....Then statement and one Goto Statement. See Program 2.

c) Write a program using an arithmetic operator and an If...Then statement. See Program 4

For .....Next

The For.....Next statement is used for counting. It can count time or counting the number of times a section of the program has been used. It is also used to repeat a set of lines (instructions) a number of times. The For.....Next statement is in the following order: For I Variable = 1 to 10. As the loop counts, I = 1, then 2, then 3.....then 10.

NOTE: EACH FOR STATEMENT MUST HAVE A NEXT TO COMPLETE THE LOOP.

1)
```
10 for I = 1 to 10
20 Print I;
30 Next I
```

NOTE: THIS PROGRAMPRINTS THE NUMBERS 1 TO 10. MAKE THIS PROGRAM PRINT Ø TO 15

2)
```
10 For J = 1 to 4
20 Print "Happy"
30 Next J
```

NOTE: HOW MANY TIMES DOES THE PROGRAM PRINT "HAPPY"

3)
```
10 Input "How many times do you wish to
print, Happy"; A
20  For K = 1 to A
30 Print "Happy";
40 Next K
```

NOTE: BY USING ANINPUT YOU CAN CHANGE THE NUMBER OF TIMES THE PROGRAM COUNTS

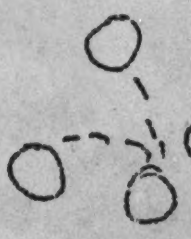
4)
```
10 Input "How long do you wish to wait?
Type a number between 1 and 3000"; A
20 Print "Hi"
30 For L = 1 to A
40 Next L
50 Print "there"
70 Goto 20
```

NOTE: TRY DIFFERENT NUMBERS TO FIND OUT HOW FAST THE COMPUTER RUNS.

B3

Gingerly

1. Print the words BUBBLE GUM on your paper.
2. Take out the L and move the E to its place.
3. Change the second letter to U.
4. Put a B after the B.
5. Drop the word GUM.
6. Put a Y for the first, third, and fourth letters.
7. Write the word to show what happens to BUBBLE GUM.



Faces

1. Copy these faces on your paper.
2. Draw a broken line from the sad face to the angry face.
3. Put a triangle around the surprised face.
4. Cross out the sad face.
5. Put an X underneath the happy face.
6. Put a small circle above the angry face

1. Read the story and the sentences.
2. Number the sentences in the order in which they happened in the story.

NAME: _____
DATE: _____
MINIMUM: 4
SCORE: _____

Mike and the Kitten

Mike and the kitten ran to the house.

"We can play here," said Mike. "We can have fun."

The kitten saw a bird. She ran to play with it. The bird did not want to play with the kitten.

"Stop! Stop!" said Mike. "Birds do not like you. It is fun for you. It is not fun for the birds. Come and play with me."

___ The kitten ran to play with the bird.

___ The kitten saw a bird.

1. Mike ran to the house.

___ "Come and play with me," Mike said.

Phi Delta Kappa, Inc., P.O. Box 789, Bloomington, Indiana 47401 SARI 50

1. Read the story and the sentences.
2. Number the sentences in the order in which they happened in the story.

NAME: _____
DATE: _____
MINIMUM: 5
SCORE: _____

Six Little Rabbits

Six little rabbits went for a walk. The little rabbits did not have a home. They saw a bird in a tree.

The bird said, "You can not come up here. Rabbits can not fly up a tree." "We will walk on," said the little rabbits.

The little rabbits saw a duck swimming in a pond.

The duck said, "You can not come here. Rabbits do not swim."

"We will walk on," said the rabbits.

The little rabbits saw a man. The man said, "You can not come here. I do not want rabbits." The little rabbits said, "We will walk on."

A mother rabbit saw the six little rabbits. "You do not have a home," Mother Rabbit said. "I like you. You may come with me."

___ The rabbits saw a man.

___ The rabbits could not fly.

___ The rabbits saw a bird in a tree.

___ Mother Rabbit liked the little rabbits.

___ A duck was in a pond.

Phi Delta Kappa, Inc., P.O. Box 789, Bloomington, Indiana 47401 SAR1 66

1. Read the story and the sentences.
2. Number the sentences in the order in which they happened in the story.

NAME: _____
DATE: _____
MINIMUM: 5
SCORE : _____

Snipp, Snapp, and Snurr

One day three little boys were talking to their mother.

Snipp said, "Mother, tomorrow will be your birthday. We want to give you something. I thought you would like a train."

Snapp said, "I thought you would like a pony."

Snurr said, "I know! You would like to have a red wagon!"

Mother thought. Then she said, "I was just wishing I could have some red shoes."

The three boys looked at one another. Then they ran into the next room to talk it over.

Snipp said, "We must get the red shoes."

Snapp said, "We do not have enough money."

Snurr said, "We must work to get the money."

So Snipp, Snapp, and Snurr ran out the door. They walked down the street looking for work.

Soon they saw a neighbor painting a fence.

Snipp said, "Do you want some help? I could paint that fence for you."

The neighbor had many other things to do. He was happy to pay Snipp to paint the fence.

Snapp and Snurr walked on, still looking for work. On the top of a house they saw a man. He was cleaning out a chimney.

Phi Delta Kappa, Inc., P.O. Box 789, Bloomington, Indiana 47401 SAR1 82

The man saw Snapp and Snurr looking up at him.

He called, "Will one of you boys help me? You are so little that you can get into the chimney. I will give you money for your work."

So Snapp started to clean the chimney.

Snurr walked on down the street. He was still looking for a way to make money. At last Snurr saw a big mill. He saw a man at the door of the mill.

The man said, "I have been wishing for a little boy to help me."

The three boys were very happy now. They had all found ways to make money. They worked very hard all day. When it was time to go home, money jingled in their pockets. Now they could buy the red shoes.

The three boys put all their money together and ran to the shoe store.

"We want to buy some red shoes for our mother," they said all together.

The man in the shoe store laughed.

"I have them here," he said. "The very best red shoes you can buy."

The boys gave the man the money and took the red shoes with them.

"Mother!" called Snipp, Snapp, and Snurr, as they ran into the house. "We have been making money," said the boys. "We wanted to buy something for your birthday, and here it is!"

Mother opened the box.

She said, "Here are just the shoes I was wishing for. They are the most beautiful shoes I ever saw! Thank you!"

_____ The boys went to buy the red shoes for Mother.

_____ Mother wanted red shoes for her birthday.

_____ The boys wanted to make money to buy the red shoes.

_____ Mother said, "The red shoes are beautiful."

_____ Snipp painted a fence. Snapp cleaned a chimney. Snurr worked in a mill.

B-4

The SARI "Systematic Approach to Reading Improvement" pre and post tests for sequencing were chosen because of the results the Co-Directors have had previously using the SARI materials. It is sequenced step by step system based on performance objectives with criterion referenced tests. The performance objectives require an accuracy from 90 to 100%.

The specific sequencing objectives seemed clearly defined and measurable. The tests were readily available and had been reliable.

The SARI kit is published by the Phi Delta Kappan Education Foundation, P. O. Box 78, Bloomington, Indiana. 47401

Sequencing Test Results

Level	No. Passed	No. Failed	When
3.3.1	11	6	pre test
4.3.1	14	3	post test
5.3.1	2	15	pre test
5.3.1	13	4	post test
	change + 11/17	- 11/17	

The first number in level is reading level. 3 is Primer, 4 is second Reader 1, 5 is second Reader 2.

The second number in the level in skill area .3 is comprehension.

The third number in the level is performance objective number..1 is sequencing.

Sequencing tests indicate an improved performance in sequencing skills tests.

Following Directions:

Four tests on following directions were administered. These were prepared by the RiverBend staff.

Results were

Test	Pass	Fail
A (pre)	11	6
B (pre)	16	1
C (post)	11	3 (3 absent)
D (post)	13	1 (3 absent)

Test results show no particular change in the students ability to follow directions. The Co-Directors felt there was too much variance in the tasks required by these tests and in another study would use a different method of evaluating following directions.

(Tests follow)

1. Read the story and the sentences.
2. Number the sentences in the order in which they happened in the story.

NAME: _____

DATE: _____

MINIMUM: 5

SCORE: _____

Lazy Jack

Jack liked to eat and sleep. He liked to sit all day in the sun. But he did not like to work.

One day his mother said, "Jack, you must find some work to do today. There is nothing to eat in the house."

Jack walked down the road. Soon he met a farmer.

"Will you let me work for you?" asked Jack.

Jack worked in his mill all day. When night came, the farmer gave him some money.

Jack started home with the money in his hand. On the way home, he lost it. When Jack told his mother about the money, she said, "Oh Jack! Do not bring money home in your hand. Put it in your pocket."

The next day Jack went to work again.

When the day's work was over, the farmer gave Jack a big, black cat.

"I will do as my mother told me," said Jack. So he put the cat in his pocket.

But the cat did not like this. She ran away. So Jack went home with nothing at all.

Jack told his mother how he had lost the cat.

She said, "Oh, Jack! You must tie a cat to a rope and let her walk in back of you."

The next day Jack went to work in the mill again. At night the farmer gave him a bag of flour to take home.

Phi Delta Kappa, Inc., P O Box 789, Bloomington, Indiana 47401 SARI 83

"I will tie this bag of flour to this rope," he said. And he did.

Jack's mother saw him come up the road pulling the flour bag. The flour was gone.

She said, "Jack! Why did you not put the bag of flour on your back?"

The next day Jack went to work for the farmer again. When the day was over, the farmer gave him a donkey.

"I will put the donkey on my back," said Jack. "That's the way to take a donkey home."

Jack looked funny walking down the road with a donkey on his back.

On the way was a big house. In this big house there lived a little girl who had never laughed.

Her father was a king. The king had said that he would give a farm to the one who could make his little girl laugh.

One day she saw Jack. He looked so funny that the girl could not help laughing.

The king was very happy. He called to Jack as he came by.

"You made my little girl laugh," said the king. "I have many farms. I will give one of them to you."

Jack and his mother and his donkey went to live on his new farm.

After this Jack worked every day. He was never called "Lazy Jack" again.

_____ He put a donkey on his back.

_____ Jack looked so funny he made the girl laugh.

_____ Jack lost the money the farmer gave him.

_____ The king gave Jack a farm.

_____ He put a cat in his pocket.

Appendix B5 Directed Essay Test

1) How do you feel about the computer instruction?
Good 9 - learned a lot - 10 like a lot-2
bad learned very little 1

2) What learned?
about computers = 13
about math = 2

3) Was it interesting?
Yes = 18 No = 0

4) Next year what would you like to learn?
More of the same again = 3 , 4 I know it all
More and different stuff 8 , 4 It takes too long

5) How much time did you spend (hours) on computer outside of the classtime?
before school 4, 10, 4, 1 = 19
noon hour 2, 4, 7, ½, ½, 4, 4, 1 = 23
after school 10, 1, 5, 1, 2, 1, 1 = 21

6) Can you think of a better way to learn about computers?
Yes : 0 No = 12

7) In your own time, what did you use the computer for?
Skill games = 6 Learn games = 3
Action games = 6
Think games = 2
Drill games = 0

8) What type of game would you like to write for the computer?
Fun = 5, A boxing match game
Fun plus sound 1 , hopscotch ,
All kinds 2 , tic tac toe, racing

COMPUTER REVIEW B-4

Correct the following programs if necessary.

1. 10 Print A="Rich"
2. 10 Print A$ = "7"
3. 10 Print BZ$ = "8"
4. 10 Print "cat"
5. "Print they went home
6. 10 Input "Type a word"; A
7. 10 Input "type a number" A$
8. 10 Input "How many runs did you make "
9. 10A=10
   20 B=2
   30
   Fill in line 30 so that the computer will print the answer 5. Use one of the arithmetic operators and the variables A and B.
10. 10 A$ = "the"
    20 B$ = "man"
    30 C$ = "came"
    40 D$ = "home."
    50
    Fill in line 50 so that this program prints the man came home. Use the string variables.
11. 10 Print "happy"
    20
    25
    30 Print "Easter"
    Fill in lines 20 and 25 with a for....next loop that will delay the printing of Easter for 100 units.

Computer Review (2)

12. 10 A = 5
    20 C = 10
    30 Print C*A
    What does the computer print when this program is run?_____

13. 10 x = 2
    20 if x = 1 Then 50
    30 if x = 2 Then 70
    40 if x = 3 Then 90
    50 Print "happy"
    60 End
    70 Print "Birthday"
    80 End
    90 Print "Sally"
    What word is printed when this program is run?_____

14. 10 Input "type yes or no"; X$
    20 If X$ = "yes" then 50
    30 If X$ = "no" Then 70
    40 Print "you did not type yes or no"
    45 Goto 10
    50 Print " I am fat"
    60 End
    70 Print "I am thin"
    What does the computer print if you type yes when this program is run?_____
    What does the computer print if you type no when the program is run?_____
    What does the computer do if you type maybe?_____
    _____
    _____

Computer Review (3)

15. Use a For.....Next loop to print the word Test 9 times.

10
20
30

16. Write a program that prints a word in each of the four fields.

10
Print "Happy",

17. Write a program that inputs two numbers then prints the answer when they are multiplied together.

10
20
30

18. Write a program that prints your name. Your age. Your school. Use only print statements.

10
20
30

19. Write a program using an input statement and an IF.....THEN statement that is different from any on this review sheet.(use back)

20. Write a program that uses the following in any order(use back)

string variable
Print
Input
If then
For...next

APPENDIX B-4 GROUP B BASIC Test

Test Total Score 35

Vocabulary Section 15 marks

Programming Section 20 marks

Scores 0 - 9 marks 14 students

12 - 13 1/2 marks 2 students

26 marks 1 student

In conferencing, the students demonstrated the ability to correct all errors due to Syntax. (A computer would indicate these as well) Programming problems were due to usually not understanding thequestion as stated, doing or solving a different problem, or confusing two questions due to where the question sentence was placed. When presented with the question orally, the child could solve the simpler problems. Three students were able to redo the test correctly. All children could answer comprehension and vocabulary "What does this program do", type questions in conference. Only the top group could sit down and write a multi-step program ready to type into the computer.

B-9

Effectiveness of Activity.

	Grade	Age	Reading	G.L	Ability Math	General Ability			Sequence				Directions				Final Test	Rank
									Pre.		Post		Pre		Post			
Name/No.	Yr.	Yrs.	Hi.	O.G.L	HI	LO	HI	MED	LO	P	F	P	F	P	F	P	F	Score	No.
A	4	10		✓	✓		✓				✓	✓		✓		✓		26	1
B	4	10	✓		✓		✓				✓	✓		✓		✓		13.5	2
C	5	11		✓		✓			✓	✓		✓		✓		✓		12	3
D	4	10		✓	✓			✓			✓	✓		✓		✓		9	4
E	4	9		✓	✓			✓			✓	✓			✓	✓		8	5
F	4	10		✓		✓		✓			✓	✓		✓		✓		8	6
G	4	10	✓		✓		✓				✓	✓		✓		✓		6.5	7
H	5	11	✓			✓		✓		✓		✓		✓		✓		6	8
I	5	13		✓		✓			✓		✓	✓			✓	✓		5	9
J	4	10		✓		✓		✓			✓		✓	✓		✓		5	10
K	5	11		✓		✓			✓		✓	✓			✓		✓	4.5	11
L	3	9		✓		✓			✓		✓		✓		✓		✓	4	12
M	4	9	✓			✓		✓			✓	✓		✓		✓		4	13
N	5	11		✓		✓		✓			✓	✓			✓	✓		4	14
O	5	10	✓			✓		✓			✓	✓		✓			✓	3	15
P	3	10		✓		✓			✓		✓		✓	✓		✓		3	16
Q	5	11	✓		✓			✓		✓		✓		✓		✓		2	17
totals					6	11	3	9	5	3	14	14	3	12	5	14	3		

Notes

1. Reading Grade Levels from Gates McGinity, May '81.

Appendix B6

Sample Anecdotal comment.

Lesson on strings (large group)
Quick response - James, Tammy, Carol,
Middle response- Chad, Jason, Leanne
Low to none - Joanne, Tricia, Jackie
Inattention - Leisa, Shayne.

Set remediation on low/middle group.

a) Review next lesson, set a homework que.
b) Assign first group as helpers
c) Extra micro block.

This would be next transferred to individual record.

Eg. Student James
810122 strings okay
810119 Print, Input - problem on Input.
810121 Input - okay

9) Any further comments.

They're fun and exciting.
I know everything now.
I first thought it was hard to handle but now it is easy.
They teach a lot, because I learned to be patient, type faster, and some computer languages.
In math it first had easy stuff then hard and that was important to me.

Computers are easy to handle.
All I can say is, I like it.

```
10 Print "What is your name?"
20 Input A$
30 If A$="teacher" then 70
40 Input "give two numbers.";A,B
50 C=A*B
60 Print "A";"times";"B";"equals";C:Goto 40
70 Print "This program is for kids"
```

quotations needed around words to be printed.

Dollar sign is used for string variables (words)

decision

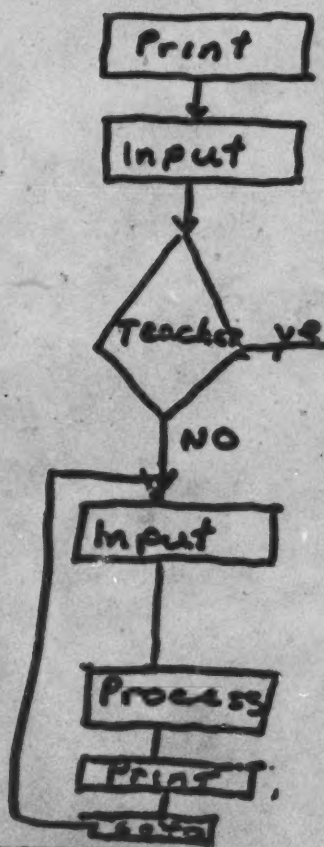
Comma between each input

Times

separates words from variables

used if more than one word

40 per line



(What is your name) is printed on the screen

Type in your name

A$ allows you to type in letters

A allows you to only type in numbers

(This program is for kids) is printed on screen

(give two numbers)

A is a variable B is a variable

A is multiplyed by B to get first inputed number (times) second inputed number (equals) C

Appendix C

WORKSHOP ON MARCH 6, 1981 -

Alberni District Professional Day.

TITLE "WHAT IS GOING ON AT RIVERBEND."

This workshop had about fifty K-7 teachers attending. The presentation was to be a video of the children on the micro, a history of the project, and some of our initial findings.

While describing the project, we found little to no background on computers amongst the teachers. The terms program, programming, hardware, software and micro computer had to be defined. The group expressed a desire to be treated just like the students - i.e. given a lesson in operating a micro computer. As a result of these needs, communicating what our project aimed to do was shelved for an introduction into computer literacy, off the cuff.

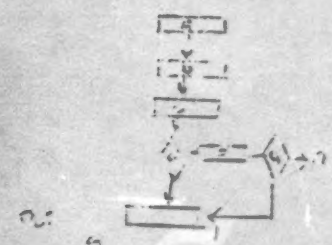
From this experience, the Directors advise future workshop givers to presample your audience. The degree of learning achieved through the popular press is not that specific. We passed out the following sheets for those wishing to implement a computer literacy program in their school. This, and the workshop were well received. Further workshops are planned for next year.

Some Suggestions for Computer Related Activities

I NO COMPUTER REQUIRED.

1. Scrap book collection of pictures and articles about computers (class project?)
2. Draw or sculpt the ultimate computer. (class needs background before this activity can happen)
3. Writing projects (creative or researched)
   a) How can computers help us.
   b) I am going to be a _____(job)_____ and knowing about computers will help me because.....
   c) Computers can create problems because.....
   d) What my parents think about computers
   e) How smart are computers?
4. Oral or debate projects see above No. 3
5. Flow charts or logic diagrams
   1. Pictures or ideas in sequence.

   example: MORNING ROUTINE



   a) Get out of bed: go to drawer
   b) Look for clothes to wear in drawer
   c) Are there any socks
   d) If there are socks put them on
   e) Or else look for socks in dryer
   f) Are there socks in dryer
   g) If there are socks in dryer put them on
   n) Or else look.....

6. RESEARCH - SCHOOL LIBRARY
   a) Parts of a computer (hardware)
   b) What makes a computer run (software)
   c) Uses of computers
   d) See Part 3 above.

7. Information from outside sources (resource center)
   1. Television shows fast forward
   2. Film strips ? Films ?
   3. Books ?

II COMPUTER REQUIRED SOFTWARE REQUIRED

1. Computer games
2. Simulations . . . . Game that preforms like real life
3. Practice . . . . example test of tables (math)
4. Speed reading . . . tacheoscope
5. Learning new information at own speed.
6. Arts
   a) drawing on the screen
   b) making designs on the screen
   c) making music on the computer.
7. Computer languages and programming logic.

III Bibliography - see attached Bibliography

An adventure in Computer Literacy

from River Bend School
6180 Compton Rd
Port Alberni, B.C.
V9Y 7B8

Staff members: Barbara Kelly, Jackie Levans, Angela Snikkers, Margaret McDonough, Reg. Nordman.

Contents

	P.
Teacher perceptions	2
Teacher preview	7
Computer terms	14
K-3 unit	16
Background material	23
Computer literacy test	47
Buying gude/ start-up	63
Glossary	71

TO ALL THOSE INTERESTED IN COMPUTER LITERACY:

This booklet was used in a small elementary school (100 students) as a teacher reference. The school was divided into four multi-age groups and led by four teachers. The teachers and groups had a common start, but they went through the twenty-four (1 hour) computer lessons in several exciting directions. All groups became involved with flow charting daily events. One group continued to evolve more and more complex charts. The historical study was knowledgeable about mathematical machines. All students made a computer book, with newspaper articles, notes on class discussions, and home interviews. The culminating activity was interaction with a TRS-80 micro-computer for all age groups.

Have fun!
Reg Nordman

INTRODUCTI ' OF COMPUTERS AT RIVER BE

ELEMENTARY SCHOOL, PORT ALBERNI

Principal: Barbara Kelly

What are microcomputers? In June 1979 I would have been quite unable to answer this question, but then I had not met Bill Goddard who was to lecture at U.B.C. that summer. Education 580 brought together a group of people some of whom were experts and some neophytes working with microcomputers. Bill was so skilful as a lecturer that we all felt we learned a great deal. Working in pairs (expert/beginner) we each wrote a teaching unit. With the help of Barry Simmons of Agassiz, I was able to complete a unit for introduction of computers in the primary grades.

September at River Bend School we had a new staff member Reg Nordman. He had worked at U.B.C. with computers and was quite willing to begin a project. Reg found a second-hand TRS 80, River Bend Parents' Association bought it, and away we went!

During the 1979-80 school year all children and staff had a "hands on" experience. Initially Reg worked the computer. Later in the year all the staff worked on the computer literacy project in Kindergarten to grade seven multiage groups. We did not structure the unit with pre-written plans, as we would Social Studies or Science units, because we wanted to explore the topic of computers and the

influence of computers on society. Each staff member's group discussions and own interests led the children to some interesting discoveries, and families appear to have enjoyed making up the computer books from newspaper and magazine articles. Flow charting was an excellent activity; it enabled the students to be creative and at the same time forced them to consider sequencing, a skill which I think is so important to success in all other subjects.

The 1980-81 school year brought a new staff member to River Bend. John Parker is very knowledgeable about microcomputers and owns a TRS 80. He and Reg Nordman have been working together since September on a computer project. They are investigating the feasibility of using the microcomputer in the elementary school classroom. Recently they were awarded an ERIBC grant for their research costs.

I am pleased by the willingness of the staff and parents' support of the introduction of computer literacy in our school. We hope soon to be byting on an Apple.

TEACHER PERCEPTIONS

COMPUTER LITERACY K - 7

by Mrs. J. Levans

This group started with what was familiar in "counting machines" - the first being the abacus. Others are adding machines and cash registers. Now there are computers that read, write, count, do arithmetic, remember things, and can follow complicated instructions.

The younger children were shown pictures of these machines while the older ones explained where these machines could be found and how they were used. For instance, who used computers?

Banks - to sort checks
Stores - to make out bills
Factories - to tell how long each person works
- how much he should be paid
- make paychecks.

This brought in the physical aspect of the computers - some children had different kinds of their own - brought them and showed how they worked. They found some were small enough to carry while others could be as large as a house.

Here the computer itself was introduced and even the Kindergarten children played simple games with it. They found in order to operate and understand they would have to know the computer terminology such as: unit, input unit, data, program, programmers, memory bank, output unit, print out.

It was then explained that in order for a computer to do the things it does, it is necessary for the programmer to give it lots of data and many instructions in a sequential pattern. In order to write a program, a programmer first makes a plan like a road map. This is sequencing. This plan is called a Flowchart.

It was interesting to find that at all levels they began to think more clearly in a logical order. They arranged themselves in groups with all grade levels in each to plan and execute a picture flowchart. The first attempt proved amusing as they had forgotten a number of steps and when pointed out to them they had presumed that those steps would have been understood. With practice, flowcharting 'flowed'. I found this exercise is the one that had atremendous influence on the Language Arts program. They learned from flowcharting that there cannot be any 'skips' in explanations nor presuming the basic understanding and knowledge of others.

C. .PUTER LITERACY AT RIVER B. D

ELEMENTARY SCHOOL.

Teacher: Angela Snikkers

As a staff member at River Bend Elementary School I became involved in a computer literacy program in the 1979-80 school year. I was not knowledgeable as to the uses of computers in schools and wondered what my part in the computer literacy program would be.

Each teacher was to have a multi-aged group of children from Kindergartem tp Grade Seven with whom we were to discuss computers and flowcharting. It was very exciting to see the interest the children had in computers. Each session consisted of a sharing of information which ranged from items in newspapers to discussions in the children's homes. I was curious as to how the children felt about computers. Did they welcome computers into their lives or were they afraid of them? What types of things could a computer perform? Would computers someday control man? Would man in fact become obselete, and if not obselete would man end up with too much leisure time and cease to be creative and imaginative? The children's feelings about these questions were mixed and each discussion sparked a new area of interest. One discussion was directed solely to the question of whether or not computers should make a decision such as choosing the next Prime Minister of Canada. Every student had something to offer to our discussion periods.

After the discussion period the children worked on their flowcharts which proved to be good sequencing exercises. The flowchart headings ranged from "Going On A Date" to "Going To School". Besides being an exercise in logic the flowcharting gave the children an opportunity to show their humour, creativity, and imagination. The children and I were very pleased with the computer session and I knew that this was only the beginning of something better to come. The coming of the next school year was anxiously awaited for everyone seemed to want to know more about computers and how they are becoming a part of our lives.

How Will a Computer Help me

The computer will help us with math and also finace → accounts. Save time, Help with taxes, -e it can keep records, Be well advanced for we can better things, For cooking you can cook things with learning it, It will help in some jobs, You can mal a computer to fit a computer. Save money, It could se people lives, Be a game., Maybe it would be afriend It could also be us as a seurvent and briwg you things and make you money

By Robbie Jackson
Gd 5

TEACHER PREVIEW - COMPUTER MONTH

A. Computer Games - demo/software
- access

B. Computer Terms

C. Flowcharting

COMPUTER MONTH

COMPUTER LITERACY INFORMATION

What is a computer?

- An automatic device used to solve problems. We have slide rules, adding machines, calculators, and a digital computer - a TRS-80.

What is a digital computer?

- An electronic device that processes information under a logic program.

What is a program?

- A logical sequence of stored instructions. Our machine uses BASIC language (Beginners All-Purpose Symbolic Instruction Code)

What are computer uses?

Accounting: payrolls, bills, tax returns,budgets

Records: sales, inventory, student records, taxes, criminals, air tickets, McDonald's sales, magazine subscriptions.

Industry: files, control assembly lines (machines) eg. sawmill at Somass, designing buildings, control processes (pulp mill).

Aerospace: navigation, systems regulation.

Science: research, lengthy math calculations, 10^6 x faster than humans analyze, weather, experiment data, simulate experiments.

Schools: teaching/learning tools, give questions, check answers, register the marks.

Recreation: create music, play games, produce art displays - large and growing area.

INSIDE THE COMPUTER

Input? receives information (people, cards, tape, other computers, T.V. screen and typewriter, discs, voice.)

Memory? storage of input, plus temporary and final results.

Control? Changes the input information into perfectly synchronized sequence of operations that processes (adds, subtracts, counts, etc.) the information.

Arithmetic Logic Unit? adds, subtracts, rearranges and makes decisions about numbers. The ALU + control form a Central Processing Unit (CPU).

Output? turns the results of processed information to the operator or another electronic device. (by TV screen or tape, discs, cards, voice, or paper printouts)

How did it begin?

1642 - Blaise Pascal, (Fr.) invents a mechanical adding and subtracting machine. A modern adaptation is a car odometer.

1671 - Gottfried Wilhelm Leibniz - designs a mechanical multiplying and dividing machine.

1834 - Charles Babbage (English) designs the advanced and complex Analytical Engine. It anticipated the modern computer, even to the point of running on programs punched on paper tape. It could not be built with the technology then available.

1936 - Konrad Zuse (Germany) designed an electronic digital computer. Built in 1941, called the Z3, it used relays to perform calculations.

1944 - IBM builds the Mark I computer for Harvard University, "The Automatic Sequence Controlled Calculator". It used gears and relays, not vacuum tubes. It was slow. It took 0.3 sec to add/subtract, 6 sec. to multiply, 11.4 sec. to divide. It was 51 ft. long, 8 ft. high, and weighed 5 tons!

1943 - 1946 - University of Pennsylvania designs the ENIAC, "Electronic Numerical Integrator and Computer", the first all electronic computer. It weighed 30 tons, used 19,000 tubes per year, required 200,000 watts of power.

1948 - The transistor is invented and starts to replace hot, power hungry vacuum tubes by 1959. The computer can now be made smaller, cheaper and more reliable.

1960's - transistor companies perfect the Integrated Circuit (IC), complete electronic circuits with a dozen or more parts on the surface of small chips of silicon. IBM produced the system 370 in the 1970's.

1965.- Digital Equipment produced the first minicomputer, the PDP-8. It was as high as a 2-drawer filing cabinet and cost 'only' $18,000.00 - a bargain.

1971 - November Intel Corp. produces the first microprocessor. A Complete CPU on a silicon chip smaller than a key on a pocket calculator. Used in calculators, microwave ovens, controlling traffic signs, to being a CPU in a personal computer. By adding memory to a microprocessor you have a microcomputer. Hundreds can be made from a large silicon crystal. But, only some of them will work. That, plus very expensive design costs meant the first microprocessors cost 100's of dollars. Now the new microprocessors cost under $10.00, and they are getting still cheaper. (Dr. Marcion E. Hoff · prop '69 # 4004 #1103)

The future holds a pocket micro-computer. Whatever happens we can still be reassured, with 12 billion neurons in a human brain, each neuron having from 5600 to 60,000 places to store input and output, all requiring very little power, man's mind will always be in charge of the computer revolution. *R.N.*

G. Hamurabi Appendix 100-a,b
Mat: TRS 80, Game tape, overhead projector.
Conc: Ratio
Skills: Prob. solve strategy, guess and test
Mental multiplication (by 2 digit)
Estimation

Math lesson

Entry- The game is loaded and running. A large monitor should by visible to all students. Let the class play with the game at the initial cantact, with no recommendations! Students usually starve their subjects.

Teacher Interrupt- Reset the game, on the o. h. projector. Draw a record column Level 1- Food distribution

Bushels # of people Ratio

Make several passes, record the food distrib. (see ex. 1.) Illustrate the ratio of bushels/person. Then have children move from a simple guess and test strategy to a guess/test/calculate ratio/strategy. As the ratio increases to 18-20:1, starvation decreases and the town attracts new settlers. Ratio becomes a very powerful tool in developing stategy. Reset and draw new record column.

Level 2

Year	Acres		Food distrib.			Plant		Store
	Buy	Tot.	Bushels	People	Ratio B:P	Acres	Yield	Bushels
1								
2								
3								

With this record system, ask the students to find ratios of total acres per person, allowable planting acres per person. As the game progresses, values could be explored. Eventually the good strategist would use the yield in bushels/acre and the value of land in bushels/acre to buy an acre to build up his fortune. What is the optimum # of residents?

Social Studies- Who is Hamurabi?
Where is Sumaria?
How do you grow wheat?
Kings and empires in the "Cradle of Civilization."
Writing/research stories about starvation, plagues, and life in those times.

N.B. The list need not end, as the game can be fitted into the curriculum of any creative teacher. Play it, then use it.

Appendix b.

	Buy	Total	Food	Ratio	People	Plant	Store
1	100	1000	700	7:1	100	0	0
2	-100	1000	100	3:1	36	72	1789
3	24	1048	96	12:1	8	24	1284
4	24	1072	96	19:1	5	24	797
5	0	1072	124	18:1	7	50	266
6	0	1072	160	20:1	8	80	438
7	0	1072	207	23:1	9	90	528
8	0	1072	286	26:1	11	110	380
9	0	1072	300	20:1	15	80	380
					16	40	199

Hamurabi

You are the exalted ruler of an ancient kingdom. Your objective is to build your kingdom from its present population of 100 poverty-stricken peasants into a wealthy, thriving metropolis.

Agriculture is the only industry, and as a wise ruler, you will decide how many acres to buy and sell at current rates (paying with bushels of wheat). Of the acres you own, you must decide how many to plant. Last year's harvest must also be distributed. Some must go to food for your subjects, some must go to seed for this year's planting, and some may be saved for future years.

There are several obstacles you will encounter. The major one is your own ignorance. Since you just inherited the throne, you don't know how much food each peasant will need to keep from starving. You have no idea how many acres each peasant can plant or even how much wheat it takes to plant an acre.

The Secretary of Commerce would not have the audacity to tell you about your mistakes; he will, however, tell you when your instructions are impossible to execute. You will have to learn by trial and error (or by looking at the program listing) what is best for the kingdom.

To resign from your position, sell all your land.

B. NOTES ON COMPUTER TERMS

Computer is used in a growing number of areas. In Alberni alone the banks, mills, telephone, hydro and soon grocery stores. (Field trips)

Computer contains units

Input unit- receives information, data, as holes in tape,cards, noise on disc/cassettes, elec. signals from keyboards, telephones, voice.
- receives info, program which is a set of instructions on what to do with the data.

What is a programmer?

Languages act as interface between the computer and programmer.
eg. BASIC, PASCAL, NATAL, APL.

Program sample

```
10   A= 7
20   B= 6
30   C= A + B
40   Print.  C;
50   STOP
```

Machine sample

```
010010    S
010011    T
010110    O
100111    P
```

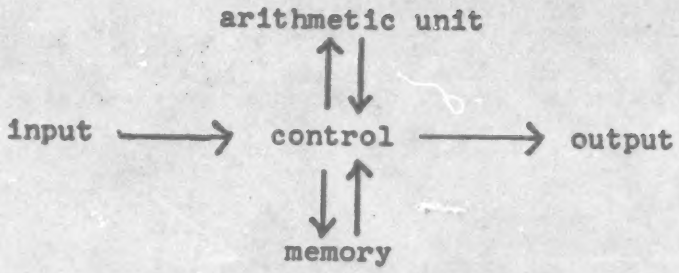
ithmetic unit- The computer is only an adding machine. The other operations are old math "tricks" which use addition!

mory unit- Uses open and closed switches to "remember",computer only remembers that which it is told to remember.

tput unit- controls print outs, T.V. screens etc. Returns the information in a form which "people" can understand.

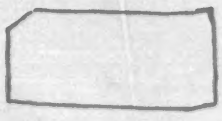
trol unit- a coordinator of information.

Computer terms continued:



Flowcharting

a map or direction giving device.
- it has its own special symbols
-The computer must have information processed according to "logic rules"
-The flowchart must be set up with those same logic rules.



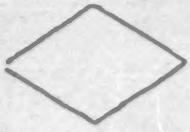
input

operation

initiolize process

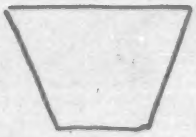
tape memory (to)

form tape



comparison decision

arrows



manual operation

T.V. display

Introduction to the Computer

Students may be introduced to the computer at the kindergarten level. The teacher should familiarize him/herself with the research papers and books listed below before embarking on computer literacy projects.

Computer Literacy Reading List

Ackerman, Judy. Computers Teach Math. The Arithmetic Teacher, May 1968.
Build a tick tack toe - learning computer

Ball, Marion J. What is a Computer? Houghton Mifflin Company, 1962.
An excellent book to read to children.

Ewing, Sam. Computer Literacy For 4th, 5th, and 6th Grades.
This paper describes the development, implementation and evaluation of a computer literacy programme for 4th-6th grade students.

Forsythe, A.L., Stansburg, Diviette H. Bobby and a Computer. The Arithmetic Teacher, February 1971.
Bobby, a third grader, learns to use a computer as a tool for solving mathematics problems.

Hollingsworth, J. Elementary Computers L'Anse Creuse Public Schools Atwood Elementary. AEDS Conference 1
Describes the use of micro computers (TRS-80's in the elementary schools.

Hopkins, Lawrence G. The Word-Problem Strand in Math Dri Using a Mini-System. AEDS Conference 1979.
Describes a newly expanded strands package for grades K-8.

Knief, Lotus M. Effects of computer based instruction an tutoring in an elementary school system. AEDS Conference 1979.
Describes the results of computer based instruction for mathematics in grades 1 - 6.

XXX

Nuffield Foundation. Computers and Young Children, Newgate Press Ltd., London E.C.I. 1972.

Rice, Jean. Experiences of "My Pal the Computer" in the primary grades. AEDS Conference 1979.
This computer literacy unit is designed to provide children in the primary grades an opportunity for exposure to the computer.

Rice, Jean. My Friend - The Computer, T.S. Denison and Company, Inc., Minneapolis, MN, 1976.
What is a computer? How are computers used? History of the computer. How does a computer work? Devices used to get information into and out of a computer.

Suydam, Marilyn N. Teachers, pupils and computer assisted instruction. The Arithmetic Teacher, March 1969.
Lists Computer Assisted Instructional Programmes for elementary school mathematics.
e.g. Accelerated program in logic for grade 4.

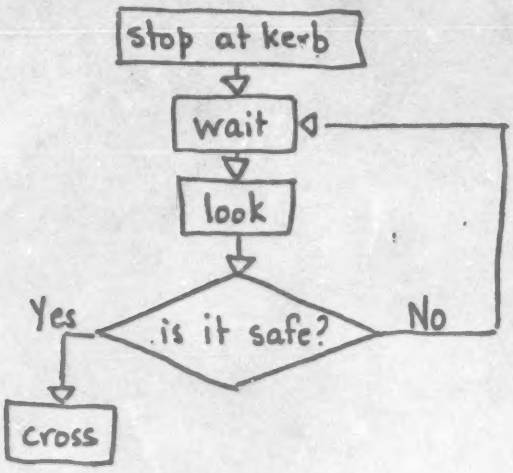
Wolfe, Lee R. Computer concepts possessed by seventh-grade children. The Arithmetic Teacher, 1968.
Suggests writing activities:
"Some of the things that computers can do for man are . . ."
"A computer 'works' something like this . . ."
"I plan to be a _____ and a knowledge of computers will be helpful because . . ."

Some ideas for teaching computer literacy in kindergarten - grade 3

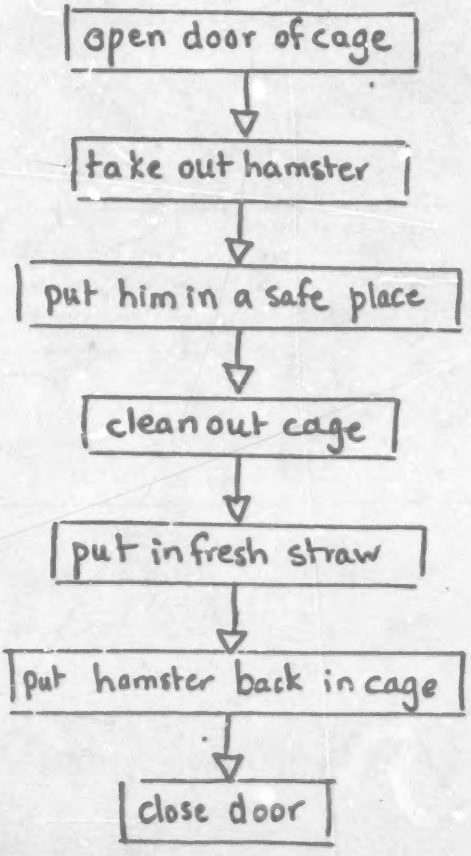
1. Build a robot.
2. Role play a robot.
3. Read "What is a Computer" to the student (see pages XXXVIII and XXXIX)
4. Flow charts - starting with picture stories in kindergar (see pages XL and XLI)
5. Practise finding PRINT, END, RUN, SCR on giant keyboard simulating a computer terminal.
6. Pretend the whole class is a computer (see XLII)
7. Make an egg box 'computer' (see XLIII
8. Type in and run a programme at a terminal (see XLV-XLVII
9. Tour a computer centre.
10. Discuss where computers are used.
11. Find newspaper and magazine articles relating to computers.
12. Use commercially produced software for drill exercises and games.

Some examples of flow charts for grades one to three

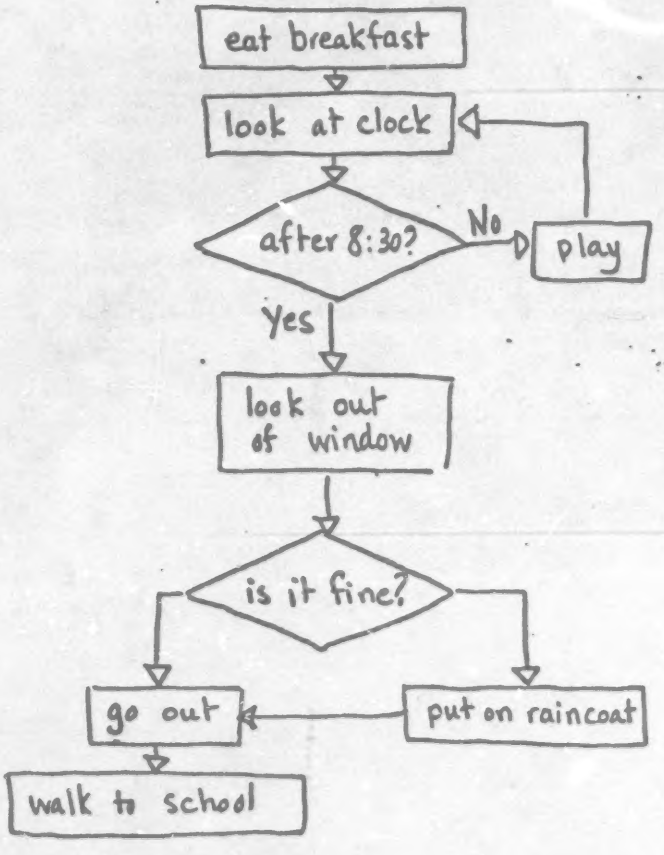
Crossing the road



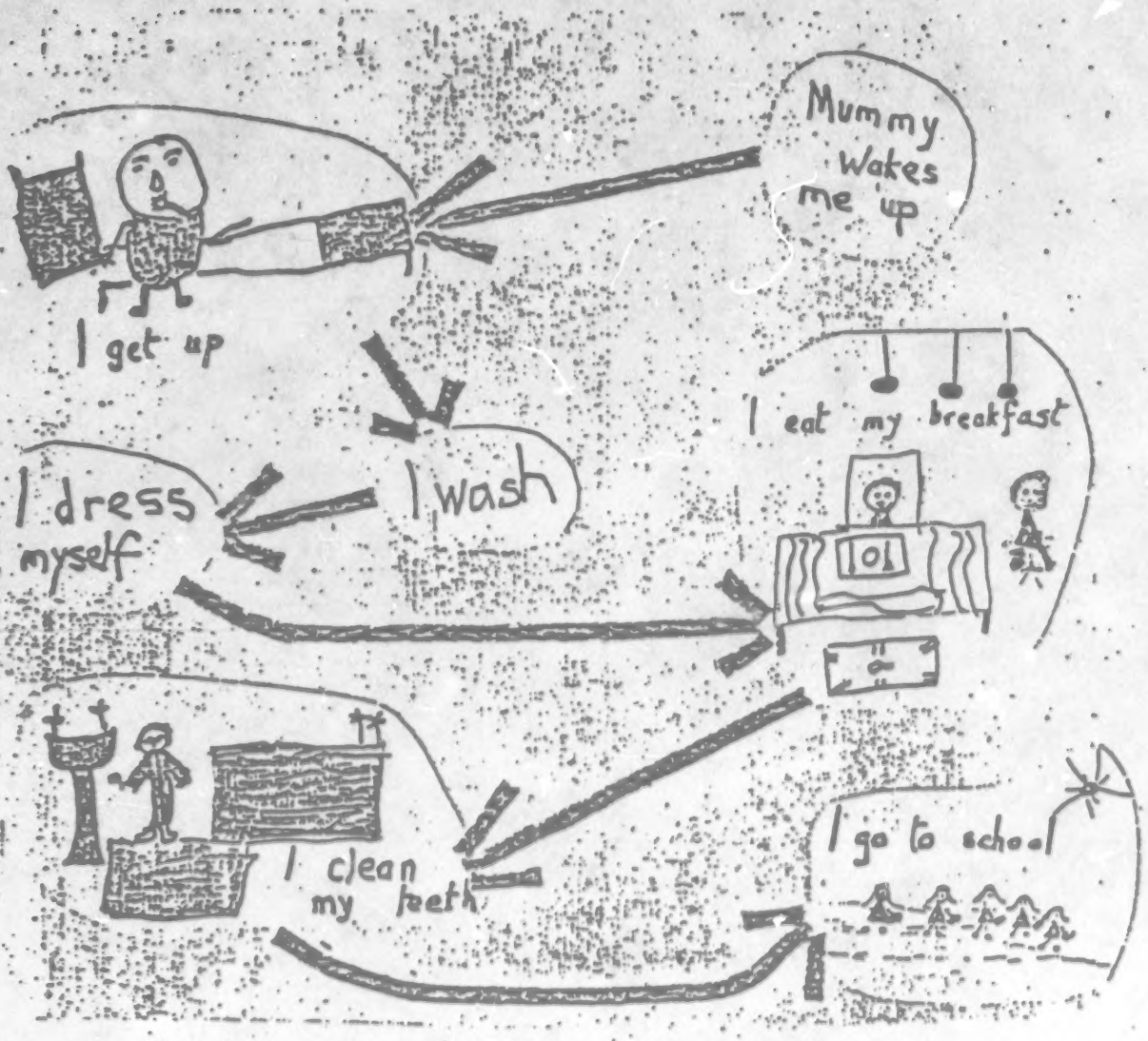
Cleaning hamster cage



Going to school



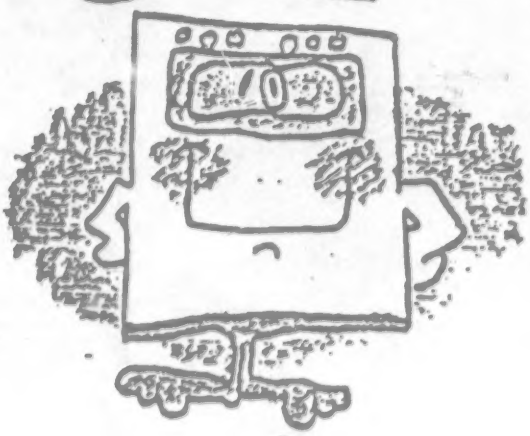
Example of a Flow Chart for kindergarten and grade one



XL

[N.B. An excellent book to read to students]

WHAT IS a COMPUTER



by Marion J. Ball

with the assistance of Nancy Levy

illustrated by Barbara A. Heisler

Conclusion to "What is a Computer"
by Marion Ball

Now that all is said and done,
we learned everything about computers under the sun

We learned about the jobs they do,
like helping in banks and libraries, too.

We learned about the computers of old,
and Herman Hollerith's story was told.

We saw generations of computers fly by,
from 1 to 4 before our eyes.

The number of parts in a computer is 5.
and all parts are needed for the answer to arrive.

Input and memory are parts one and two.
Information goes in and is stored there for you.

Control and arithmetic units are next,
and there's one more part, as you would expect.

The output equipment is where answers come out,
and that's what computer HARDWARE'S about!!

The computer's SOFTWARE as we know,
is the program that makes the computer go.

We learned that a flow chart is a little map,
where we list our whole problem, step by step.

Last but not least is the programming, it's true.
The programming unlocks the computer's power for you.

Kelly

COMPUTER LITERACY QUESTIONNAIRE

MINNESOTA EDUCATIONAL COMPUTING CONSORTIUM
2520 Broadway Drive
St. Paul, Mn. 55113

Supported by the National Science Foundation. Grant No. SED77-18658

Your Name (please print)_____

Dear Student:

We would appreciate your help by having you answer the questions in this booklet. There are two parts: the first part asks for your opinions and attitudes and the second part is a test of your knowledge about computers. Keep in mind that in the first part (the attitude questions) there are no right answers or wrong answers; just select the answer that best expresses how you feel. In the second part (the computer knowledge test) there may be items you have not yet learned. Just answer as many as you can. Keep in mind that the right answer is the best choice for each question.

Thank You.

This questionnaire developed by:

Ronald E. Anderson
David C. Johnson
Thomas P. Hansen
Daniel L. Klassen

PART 1

DIRECTIONS: Indicate how much you AGREE or DISAGREE with each of the following statements by circling the appropriate letter. Circle "a" if you STRONGLY DISAGREE with the statement. Circle "b" if you DISAGREE with the statement a little. Circle "c" if you are UNDECIDED about whether you agree or disagree with the statement. Circle "d" if you AGREE with the statement a little. Circle "e" if you STRONGLY AGREE with the statement.

As an example, if you AGREE a little that computers are noisy, then circle "d" as shown below:

Computers are noisy a b c (d) e

Or, if you are UNDECIDED about whether computers are noisy, circle "c" as shown below:

Computers are noisy a b (c) d e

If you have any questions, ask your teacher.

		STRONGLY DISAGREE a	DISAGREE b	UNDECIDED c	AGREE d	STRONGLY AGREE e
1.	I would like to learn more about computers..	a	b	c	d	e
2.	Working with a computer would probably make me feel uneasy or tense.....	a	b	c	d	e
3.	I feel helpless around a computer.....	a	b	c	d	e
4.	Computers sometimes scare me.....	a	b	c	d	e
5.	I would very much like to have my own computer.....	a	b	c	d	e
6.	I would like the idea of taking computer courses.....	a	b	c	d	e
7.	I enjoy using computers in my classes.....	a	b	c	d	e
8.	Walking through a room filled with computers would make me feel uneasy.....	a	b	c	d	e

	STRONGLY DISAGREE	DISAGREE	UNDECIDED	AGREE	STRONGLY AGREE
	a	b	c	d	e
9. I feel uneasy when I am with people who are talking about computers.....	a	b	c	d	e
10. I enjoy working with computers.....	a	b	c	d	e
11. I feel confident about my ability to use computers.....	a	b	c	d	e
12. It is my guess that I am <u>not</u> the kind of person who works well with computers.....	a	b	c	d	e
13. On the whole, I can cope with computers in my daily living.....	a	b	c	d	e
14. I am able to work with computers as well as most others my age.....	a	b	c	d	e
15. Computers are gaining too much control over people's lives.....	a	b	c	d	e
16. In general, females can do just as well as males in computer careers.....	a	b	c	d	e
17. More females than males have the ability to become computer specialists.....	a	b	c	d	e
18. Using computers is more for males than for females.....	a	b	c	d	e
19. Studying about computers is just as important for females as for males.....	a	b	c	d	e
20. Men make better scientists and engineers than women do.....	a	b	c	d	e
21. Falsifying information in computers is a serious crime.....	a	b	c	d	e
22. Access to personal information in computer files is a serious problem.....	a	b	c	d	e
23. Organizations should <u>not</u> be allowed to create secret computer files containing detailed information regarding people's personal lives.....	a	b	c	d	e
24. Because of computerized information files, too many people have information about other people.....	a	b	c	d	e

57

	STRONGLY DISAGREE	DISAGREE	UNDECIDED	AGREE	STRONGLY AGREE
	a	b	c	d	e
25. To protect people's privacy it is necessary to have laws regarding computer files that contain personal data.....	a	b	c	d	e
26. Every secondary school student should have some minimal understanding of computers.....	a	b	c	d	e
27. Every secondary school student should be able to write a simple program.....	a	b	c	d	e
28. Every secondary school student should learn about the role that computers play in our society.....	a	b	c	d	e
29. Computers can be a useful instructional aid in many subject areas other than mathematics.....	a	b	c	d	e
30. Computers provide more disadvantages than advantages in education.....	a	b	c	d	e

DIRECTIONS: Indicate whether you think each of the following values is UNIMPORTANT, IMPORTANT, or EXTREMELY IMPORTANT by circling the appropriate letter. Circle "a" if you think the value is UNIMPORTANT. Circle "b" if you think the value is IMPORTANT. Circle "c" if you think it is EXTREMELY IMPORTANT.

As an example, if you think saving money is EXTREMELY IMPORTANT, circle "c" as shown below:

Saving money a b (c)

	UNIMPORTANT	IMPORTANT	EXTREMELY IMPORTANT
	e	b	c
31. Freedom	a	b	c
32. World Peace.....	a	b	c
33. Economic Growth.....	a	b	c

52

	UNIMPORTANT a	IMPORTANT b	EXTREMELY IMPORTANT c
34. Scientific Knowledge.....	a	b	c
35. Privacy.....	a	b	c
36. Technological Advancement.....	a	b	c
37. Computerization.....	a	b	c
38. Efficiency.....	a	b	c
39. Love and Friendship.....	a	b	c
40. Self Respect.....	a	b	c

DIRECTIONS: Below are some adjectives that can be used to describe computers. For each adjective circle the alternative which best expresses how you feel about computers. If you aren't sure how you feel, circle "undecided."
As an example, if you feel that computers are very big, then circle as shown here:

a. not big b. big (c. very big) d. undecided

If you feel that computers are not big, then circle as shown here:

(a. not big) b. big c. very big d. undecided

Circle one alternative for each of the eight adjectives.

COMPUTERS ARE:

41. a. not personal b. personal c. very personal d. undecided
42. a. not frustrating b. frustrating c. very frustrating d. undecided
43. a. not good b. good c. very good d. undecided
44. a. not humanizing b. humanizing c. very humanizing d. undecided
45. a. not challenging b. challenging c. very challenging d. undecided
46. a. not bad b. bad c. very bad d. undecided
47. a. not impersonal b. impersonal c. very impersonal d. undecided
48. a. not dehumanizing b. dehumanizing c. very dehumanizing d. undecided

PART II

DIRECTIONS: For each of the following questions, circle the letter beside the best answer. If you do not know the answer to a question, do not leave the item blank; circle the letter beside "I don't know." Use the "I don't know" response as little as possible. Use the "I don't know" response only when you don't even have a guess about the best answer. Do NOT leave any item blank that you attempt; either circle the letter beside an answer or "I don't know."

1. Police sometimes use computers to help identify stolen cars.

   a. true
   b. false
   c. I don't know

2. Most hospitals give injections by computer.

   a. true
   b. false
   c. I don't know

3. Computers cannot be used to assist in teaching English grammar.

   a. true
   b. false
   c. I don't know

4. Computers are not really used very much yet except by scientists.

   a. true
   b. false
   c. I don't know

5. Government officials use computers to store and retrieve large amounts of information about citizens.

   a. true
   b. false
   c. I don't know

6. People often use computers to store large amounts of information they wish to use over and over again.

   a. true
   b. false
   c. I don't know

7. Computers help people make decisions by providing correct answers to any question.

   a. true
   b. false
   c. I don't know

54

8. Computers help people make decisions by telling them if their problem is important.

   a. true
   b. false
   c. I don't know

9. Computers have been used to make more information and products available to the consumer.

   a. true
   b. false
   c. I don't know

10. Computers are used to commit crimes, especially stealing money and stealing or falsifying information.

    a. true
    b. false
    c. I don't know

11. Identification numbers and passwords are a primary means for restricting undesired access to computer files.

    a. true
    b. false
    c. I don't know

12. Use of computers in education always results in less personal treatment of students.

    a. true
    b. false
    c. I don't know

13. Privacy is an issue with files containing personal information about people.

    a. true
    b. false
    c. I don't know

14. The increased use of computers in our society both eliminates and creates jobs.

    a. true
    b. false
    c. I don't know

15. Almost all people in our society are affected in some way by computers.

    a. true
    b. false
    c. I don't know

16. In order to use a computer you would have to be in the same building as the computer.

a. true
b. false
c. I don't know

17. Computers are able to think in every way just like people.

a. true
b. false
c. I don't know

18. Using computers can free one to do more creative tasks, but this may lead to more dependence upon machines.

a. true
b. false
c. I don't know

19. In order to use any computer you would have to use a telephone.

a. true
b. false
c. I don't know

20. In order to use a computer a person must know how to program.

a. true
b. false
c. I don't know

21. Computers are <u>not</u> good for tasks that require:

a. speed
b. accuracy
c. intuition
d. something to be done over and over again
e. I don't know

22. If your charge account bill has an error, it was probably caused by:

a. breakdown of the computer
b. mistakes made by people
c. poor design of the computer
d. general weaknesses of machines
e. I don't know

23. The main duty of a computer programmer is to:

a. operate a computer
b. prepare instructions for a computer
c. schedule jobs for a computer
d. design computers
e. I don't know

24. The computer related job closest to that of a typist is:

a. computer operator
b. keypunch operator
c. systems analyst
d. computer programmer
e. I don't know

25. Which of the following persons is the most likely to be associated with the design of computers?

a. keypunch operator
b. computer operator
c. computer programmer
d. computer scientist
e. I don't know

26. A basic use of computers in libraries involves:

a. information storage and retrieval
b. simulation and modelling
c. process control
d. computation
e. I don't know

27. A basic use for computers in the design of airplanes is:

a. simulation and modelling
b. process control
c. making reservations
d. keeping inventory
e. I don't know

28. The most questionable use of large computer files is:

a. government planning
b. research
c. checking on people
d. administration of social programs
e. I don't know

29. Which of the following is a limiting consideration for using computers?

a. cost
b. software availability
c. storage capacity
d. all of the above
e. I don't know

30. Which is <u>not</u> a characteristic of most information systems?

a. a large volume of information is stored and used
b. the information is organized
c. the basic purpose is to provide reports and summaries of the data
d. they contain only alphabetic data
e. I don't know

31. The decade of first extensive manufacturing of computers was:

a. 1860's
b. 1890's
c. 1920's
d. 1950's
e. I don't know

32. Computer software is a term describing:

a. computer programs
b. electronic components encased in soft plastic or rubber
c. people who work with computers
d. mechanical and electronic parts of a computer system
e. I don't know

33. In addition to input and output equipment, computers contain:

a. terminals, paper, transistors
b. memory units, control units, arithmetic units
c. printers and typewriters
d. telephones, keyboards, television screens
e. I don't know

34. A computer <u>system</u> is best described as:

a. processing
b. programming, input, and output
c. input and output
d. input, processing, and output
e. I don't know

35. The physical parts of a computer are referred to as:

a. programs
b. hardware
c. software
d. manuals
e. I don't know

36. When in operation, a computer:

a. follows a set of instructions written by people
b. thinks just like a person
c. recalls answers from memory
d. translates data from digital to analog code
e. I don't know

37. Computers <u>cannot</u> run without:

a. blinking lights
b. keyboards
c. instructions
d. all of the above
e. I don't know

38. In order to program a computer, a person:

a. can use any English language words
b. can use any English or foreign language words
c. must use programming language numbers, not words
d. must use the words from a programming language
e. I don't know

39. At any given moment, a computer's memory unit can store:

a. programs
b. data
c. answers
d. all of the above
e. I don't know

40. Data processing is best described as:

a. the collection of data
b. producing reports
c. manipulating data according to instructions
d. using punched cards in a keypunch machine
e. I don't know

41. A computer program is a:

a. course on computers
b. set of instructions to control the computer
c. computer generated presentation
d. piece of computer hardware
e. I don't know

42. Computer processing of data may involve:

a. searching
b. summarizing
c. deleting
d. all of the above
e. I don't know

43. The computer must have two types of information to solve a problem:

a. the problem and the answer
b. the name of the program and user number
c. the data and the instructions
d. the name of the program and your name
e. I don't know

44. A newspaper publisher has the following information about subscribers stored in the computer. They are name, address and renewal date. How would you arrange the information to be most useful to the delivery person?

a. ordered listing by address
b. ordered listing by renewal dates
c. alphabetical listing of streets
d. ordered listing by zip code
e. I don't know

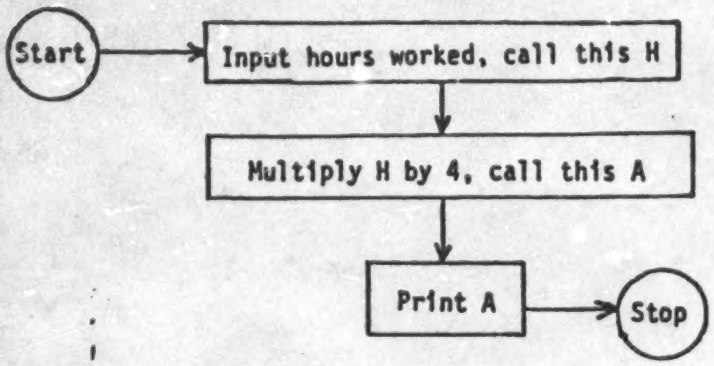
45. Choose the correct output for the procedure described below:

1. List the three names Brown, Anderson and Crane in alphabetical order
2. Remove the last name from the list
3. If only one name is left, stop. Otherwise, go on to step 4
4. List the remaining names in reverse order
5. Go back to step 2

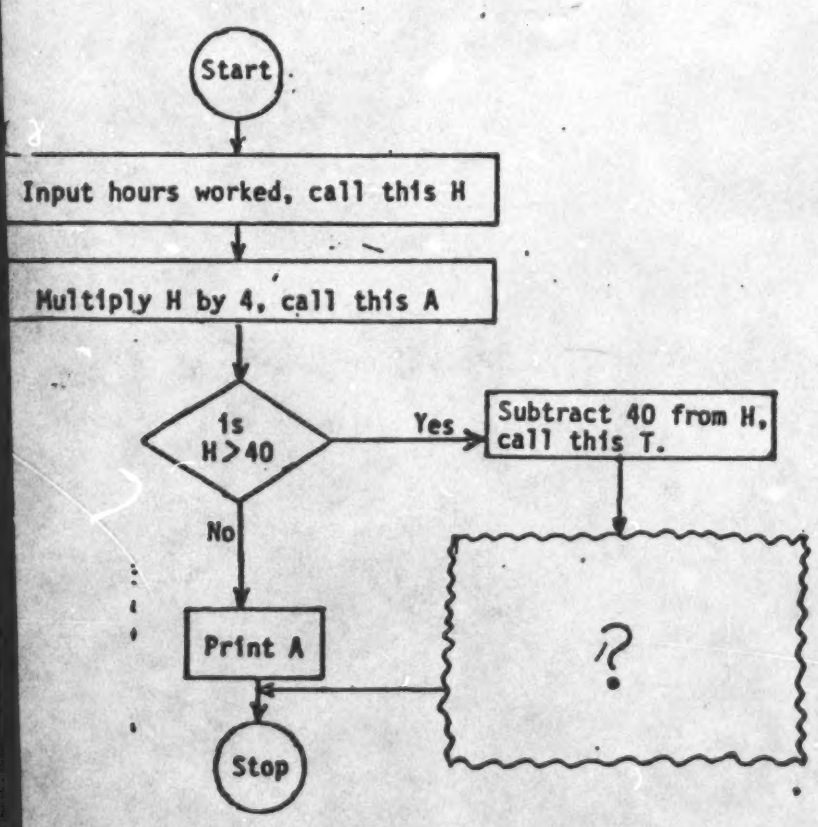
Output

a. Anderson, Brown, Crane
b. Brown
c. Anderson, Brown
d. Anderson
e. NONE OF THESE.

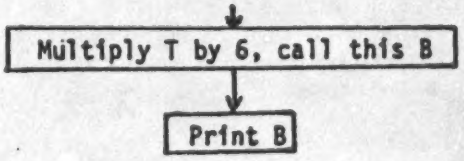
46. An algorithm (flowchart) to determine the weekly wages of employees in a bakery is shown below. Employees are paid $4 per hour up to 40 hours per week.



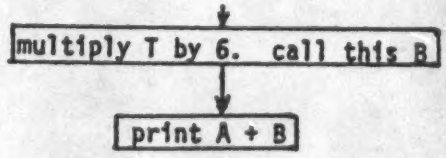
Employees are also paid "time-and-a-half" ($6 per hour) for overtime (hours worked over 40). How would you extend the flowchart below to include overtime pay. Select answer a, b, c, d, or e.



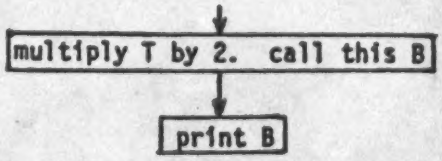
a.)



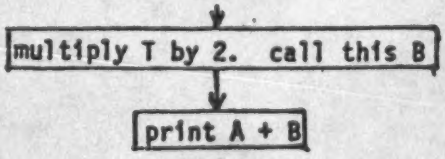
b.)



c.)



d.)



e.) I don't know

DIRECTIONS: If you have never written a computer program, STOP: do NOT answer items 1 - 3 below. Answer items 1 - 3 only if you have written a computer program before.

1. Choose the correct output for the computer program shown below:

```
1       LET A = 3
2       LET B = 4
3       LET C = A
4       LET B = C
5       LET A = B
6       PRINT A,B
7       END
```

Output

a. 3 4
b. 4 3
c. 3 3
d. 4 4
e. I don't know

2. When run on a computer, the following program will:

```
1       INPUT A, B, C, D, E
2       LET S = A+B+C+D+E
3       LET M = S/5
4       PRINT S,M
5       END
```

a. Calculate the sum of five input values
b. Calculate the average of five input values
c. Print the sum and average of five input values
d. all of the above
e. I don't know

3. This program instructs the computer to count by two.

```
10  LET M = 0
20  LET M = M + 2
30  PRINT M
40  IF M<100 THEN 20
50  END
```

Which change will produce a program which can be used to count by A? (For example, A=3, 5, or 8.)

a.
```
5 READ A
7 DATA 3,5,8
```

b.
```
5 LET M = A
30 PRINT A
```

c.
```
5 INPUT A
20 LET M = M + A
```

d.
```
5 LET X = A
20 LET M = X + A
```

e. I don't know

COMPUTERBUYIN

THE choice of computers spread before you in the pages that follow may seem dizzying. But choosing the right one isn't all that hard once you know what you want it for and what requirements your need imposes.

So the logical starting place is the application. Microcomputers are used for a wide variety of purposes: for program development and teaching oneself to program; for business applications such as accounting and inventory; for word processing to develop cleanly typed reports, letters and the like; for small mass mailings; for education in non-computer subjects; to handle home data such as recipes, Christmas card lists and the checkbook; to control home and industrial devices; to play games; and for mathematical computation.

The more such applications you have, the more sense computers (basically all-purpose devices) make. For some single applications, in fact, alternatives to the computer make more sense. If all you want is to play games, for instance, get a programmable video game, and be done with it. The game will probably cost less, and put more interesting, cartoon-like graphics on your TV screen. Similarly, if all you need to do is complex calculation, consider a programmable calculator. Again, the cost will be less—and you'll be able to carry the calculator with you at all times.

But calculators can play only limited games, and TV games have only limited calculating ability, if any (not counting the small but growing number of games that can be converted into full-fledged computers). If you're interested in both these applications at once--or in any of the others so far mentioned—you'll need a full-fledged computer.

But which one? All computers have some similarities: They all have some sort of *input* device to enter programs and data, some sort of *output* device to verify the input data and show what results the computer comes up with when the program runs. They all have *processors*, the chips that do the actual computing; and *memory* to hold programs and data while they're being used. But the types of input, output and processor differ, as do the amount of memory and the number of accessories or *peripherals* which can be used with the system.

Input and Output. The most visible differences between computer systems are usually in their input and output (I/O, for short) facilities. These are channels of communications between the computer and you. Each speaks very different languages, and one measure of I/O sophistication is how cleverly the system can disguise that fact.

In its most primitive (and, today, rarest) form, the system will communicate in *binary*, a numbering system based on twos. A completely binary I/O system would have a row of eight switches to input each 8-bit computer command or data "word" and eight lights per "word" for output.

More commonly, the system will translate such binary numbers as "11000000" into an octal (base-8) number such as "300" or a hexidecimal (base-16) number such as "C0". (Since hex numbering requires more digits than our base-10 decimal system, it follows the digits 0-9 with the letters A-F.) Many low-priced, single-board computers have calculator-like keypads and displays for either octal or hex input and output.

But octal and hex are only more sophisticated ways of talking *machine language*, the instructions that computers understand directly. Machine-language programs run very quickly, and don't use much memory. But they're cumbersome to write since you must not only learn at least a hundred or so instructions and how to use them, but must learn them as abstract numbers like "CD" or "305".

Consequently, keypad-and-display computers are only useful as is, for writing very short programs, especially programs designed to interact with other devices rather than with people. Control applications are often a perfect match for these computers. Here, the limitations of keypad programming aren't serious, and the computers are small and cheap enough to be assigned to specific devices, or sometimes even to be built into them.

But most such computers also have

TIPS

Step-by-step guide to selecting a microcomputer system

Note: For computer terminology, see Computer Glossary, page 98. Microcomputer Products Directory starts on page 100.

BY IVAN BERGER

ports for communicating with other I/O devices. Connect one to a *terminal*, which combines a full typewriter-like keyboard with a video display screen or a printer, and you can work with other programming languages which use the entire alphabet and other symbols.

With the keyboard's full set of characters at your command, you can program in assembly or high-level languages. *Assembly* language is just a word-for-word translation of machine language from abstract numbers into more easily memorized abbreviations. In 8080 assembler, for example, the instruction "return if not zero" is "RNZ". In machine language, it would be either "C0" (hex), "300" (octal) or "11000000" (binary). A program called an *assembler* translates the mnemonic abbreviations into machine code, as well as performing such useful tricks as letting you call subroutines (frequently invoked sub-programs) by name, instead of remembering their memory addresses.

But that's still doing things the computer's way, not yours. *High-level languages*, such as BASIC or PASCAL, use standard English words (though sometimes in abbreviated form) to represent whole sequences of computer operations. In BASIC, for example, "PRINT SOR(SIN(Y))" will make the computer tell you what the square root of the sine of Y is. An assembly-language program for that would probably fill up this column.

Just as with assembly language, a special program is needed to translate your BASIC or other high-level language program into the computer's commands. That program can be read into the computer from a tape, or can be permanently built into the computer's memory. If you use BASIC a lot, it is a great convenience to have it instantly on tap whenever you turn the computer on. If you don't, this feature won't make much difference to you.

The typewriter keyboard and video screen are the most common microcomputer I/O devices, but there are variations and alternatives available. Many of these systems let you not only display letters and numbers (*alphanumerics*) on the screen, but "draw" pictures (*graphics*) on the screen as well. The pictures are often rather crude, being composed of clearly noticeable blocks, but they're useful for such applications as games, graphing mathematical functions, and in business for bar-graph and other displays that are easier to understand than tables of numbers. Color makes the games more exciting and the bar-graphs more readable, but raises the cost of the computer, too.

Graphics programs written in BASIC run very slowly; for speed, you'll have to use assembly language programs. Bear that in mind if you plan to write your own graphics. If you want fast graphics at low cost, you'll find a few graphics-capable machines with hex keypad input for machine-language programming.

Even alphanumeric video displays differ. Some computers have built-in video monitor screens. Others are usually sold with a video screen in a separate cabinet. Still others include video output circuits to feed signals to a video monitor screen. To feed it to a regular TV receiver, though, you'll have to convert that signal to a modulated radio-frequency one by passing it either through an *r-f modulator* or through a video-cassette recorder, if you have one. Not all computer/recorder combinations work well, though, nor do all r-f modulators. (The latter cannot be legally sold unless it's in kit form.) Try to check out your combination in the store or on a money-back guarantee. Computers with built-in r-f modulators are beginning to appear, too. This feature makes most sense in home systems, where there's likely to be a TV receiver available, than in a business or industrial system.

There are also differences in how much information you can put on the video screen. Alphanumeric displays are available with 16 lines of 64 characters each, or less, and with 25 lines of 80 characters, or more. Graphics displays also differ in the number of horizontal and vertical elements they can show—the amount of picture detail, in other words. The more information you pack on one screen, the more you take in at one glance. But more detailed displays cost more, and require higher-resolution monitors. As a result, high-density displays often cannot be used with r-f modulators and TV receivers.

Keyboards are more standardized. The basic differences are in keyboard "feel" (more likely to matter to an operator who already knows touch typing than to a hunt-and-peck operator) and in the presence or absence of separate numeric keypads. These keypads are very worthwhile in applications involving large amounts of numerical entries, such as in business accounting or in scientific computation. It's far quicker to punch numbers into a calculator-like nest of keys in a compact bunch than to use a row of number keys spread out across the top of the keyboard.

One major difference between typewriters and computers is that some computers display only upper-case, or capital letters. That's fine for most applications, but not for word processing.

Word processing systems are mostly used for business, where it costs lots of money to turn roughly typed or written drafts into smoothly typed letters and reports. On a typical word-processing system, the operator can enter text, make corrections of all kinds, then command the computer to print out a perfectly typed, finished copy. If it's a form letter, the computer can turn out a separate copy for each name and address on its list. Such systems are being adopted by offices, by free-lance writers and others.

Most small computers communicate with you through video screens. For most applications, this makes perfect

sense. video systems are fast, silent, reliable, and don't use up paper.

But there are times when it definitely pays to have a permanent record of the computer's output. Word processing is an obvious example, but so are accounting (including your personal checkbook), alphabetizing of lists, or making written records of your programs that you can send to friends or carry with you while you look for problems and improvements. Properly programmed, a computer could print out your shopping list in the order that the items appear in on your supermarket's shelves.

In the early days of small computers, Teletype® printing terminals were the most common I/O devices. Today, video screens—on terminals or connected directly to the computer—are. But most systems do allow separate printers to be added to the system. If this is important to you, check how easily the printer can be added to any system you're considering, and how much the printer and its connections will cost.

Inside the Computer. It's no accident that we've been talking only about externals so far. For the input-output communication channels between you and the computer have far more to do with its utility than many of the circuits inside do.

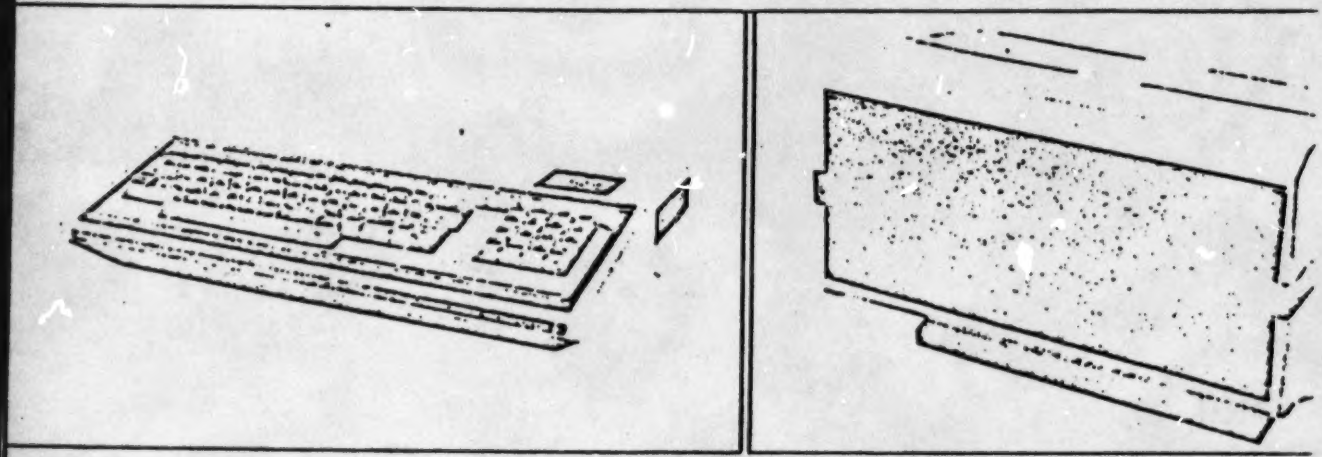
The most important of these circuits is probably *memory*. You'll find computers here with as few as 256 "bytes" of memory, each *byte* being an 8-bit computer "word" that can represent a single alphanumeric symbol or a single computer command. You'll also find that many are expandable to as many as 65,536 bytes, variously abbreviated as either "64K" or "65K". (The two figures are equivalent; the "K" stands for "1024", a binary number that's only a little different from the decimal 1000, usually abbreviated "k". The figure of 65,536, representing 64 of those "binary thousands", could be abbreviated as either 64K or, in decimal, 65k; but the capital "K" is used, confusingly, for both, in this one instance.)

Most systems, though, fall into the 2K to 32K range. Memory costs money, so the more you have, the more the system costs. But the more memory you have, the longer the programs you can store, and the more data you can have available for them to work on.

There are two types of memory. RA and ROM. *RAM* (Random-Access Mem ory) is used for temporary storage programs and data and for the resul of program runs. The contents of RA can be changed at will, and many them change constantly during the ru ning of a program. But those conten also fade out within seconds when it power is turned off.

That's where *ROM* comes in. RO (Read-Only Memory) doesn't forget but you can't readily change it, eithe Hence, ROM is used to hold vital pr grams which you'll use all the tim such as those which instruct the cor puter how to accept input from the ke board. Some computers have BASIC ROM, too—on others, you have to loa in the BASIC language program from tape each time you use it.

Most computers have more RAM th ROM. Typically, a system will wind having about 2K of ROM (about 8K 10K with BASIC in ROM) and 16K more of RAM (less, if BASIC is in RO since that frees up the RAM space th BASIC would otherwise occupy). Th may start with less, but sooner or late more memory is added.



For A Demonstration Or Further Information Contact Your Local Computer Store.

Exidy inc.

#1

Some inexpensive systems, usually
e single board, keypad and display
have very limited RAM space on
, (perhaps 1K or 2K). Most of
se allow other boards to be connect-
h more RAM. But unless your ap-
n is a simple one using machine-
guage or assembly-language pro-
ms (device control, for example), be
any system you buy can be ex-
ded to include enough memory for
our needs. There's no hard-and-fast
about how much is enough, except
business systems seem never to
enough memory.

Storage. Programs, other than
in ROM, must be fed into the com-
every time you turn the system on
itch from one program to another.
ng them each time from the key-
or keypad is ridiculously time-
ming, and almost inevitably leads
ors. So it's vital to have some
foolproof way to save programs
enter them.

use of punched paper tape has
died out, since it's a slow and
rocedure. Most small computer
standardize instead on cas-
sette tape, either built in or as an accessory program storage device. Most such systems convert programs and data into tones which can be recorded on ordinary audio cassette recorders, but a few record digital pulses, not audio tones, which requires a special recorder. Cassettes, especially audio cassette systems, are fairly slow (they require several minutes to load BASIC, for example). But they're faster than paper tape, use tape you can buy almost anywhere, and they usually make extra use of a cassette recorder you already own. Cassette programs are not always interchangeable between different computer makes, though a few cassette formats available as accessories for many computers, have achieved fairly wide use.

If you need more reliable loading (cassettes sometimes have to be loaded several times before you get them right), quicker loading, and faster access to a wide variety of programs and data, then it's time to consider *floppy disks*. Floppies are basically magnetic recording tape cut into discs instead of ribbons. They use digital recording and are very fast—BASIC or other long programs typically load in seconds. They also speed up access to programs and data. Getting from the first program on the disk to the last is a matter of moving the head a few inches from the outside to the inside track. In contrast, getting from the first to the last program on a C-60 cassette means moving about 250 *feet* of tape past the head. Disks also load more reliably than cassettes. And, when they don't load properly, reloading takes only a few seconds more.

Unlike cassettes, disks allow much interchangeability between computer systems, especially with systems based on the 8080, 8085 or Z80 processors. Many companies sell 5¼-inch disk programs written for use with these systems. Moreover, Digital Research's CP/M operating system simplifies interchange of programs from different computers using the foregoing processors.

The *processor* is, for the most part, less important than the system you use it in. If you're programming in BASIC or some other language, you'll find as much difference between versions of BASIC running on a common processor as between versions running on altogether different ones. If you program in

	EXIDY SORCERER	TEXAS INSTRUMENTS 99/4	ATARI 800	APPLE II	COMPUCOLOR MOD III	COMMODORE PET	TANDY TRS-80
	$995	$1150	$999.95	$1150	$1495	$795	$599
	Z80	9900	6502	6502	8080	6502	Z80
	48K	16K	48K	48K	32K	8K	16K
	12K	26K	16K	8K	17K	14K	4K
	B/W	Color	Color	Color	Color	B/W	B/W
	64	32	40	40	64	40	64/32
	30	24	24	24	16/32	25	16
	512/240	192/256	384/192	280/192	128/128	320/200	128/48
	79 Key Typewriter	40 Key Calculator	57 Key Typewriter	52 Key Typewriter	77 Key Typewriter	73 Key Calculator	53 Key Typewriter
	Yes	No	No	No	No	No	No
	Yes	No	No	No	Yes	Yes	No
	128	No	No	No	No	No	No
	Dual Cassette RS232 Communications 8 Bit Parallel	Joystick Sound	Joystick Serial Single Cassette	Single Cassette Joystick	Single Disk RS232 Communication	Single Cassette IEEE 488	Single Cassette
	S-100	No	No	Yes	Yes	IEEE 488 Daisy Chain	Yes
	630K Byte	No	92K Byte	116K Byte	51.2K Byte	125K Byte	45K Byte
	ROM Basic ROM Assembler ROM Word Processor CPM EXT. Basic CPM Fortran CPM Cobol CPM APL CPM Pascal	ROM Basic	ROM Basic ROM Assembler	ROM Basic Disk Basic Pascal	Disk Basic	ROM Basic Disk Basic	ROM Basic Disk Basic Cassette Assembler

...ilable June 1979

390 Java Ave. Sunnyvale CA. 94086 (408) 734-9410

CIRCLE NO 8 ON FREE INFORMATION CARD

assembly or machine language, you'll find unlike processors very different to work with, but you'll also find that every processor has its firm adherents, with each processor's advantages being balanced by disadvantages relative to other processors. The best way to choose is to settle for whatever processor is in the system which best suits you, and for which the programs you need are already available.

There are, however, some exceptions. To begin with, there are now several processors (such as the TI 9900 and the LSI-11) which use 16-bit "words" instead of the 8-bit ones used by most microprocessors. This allows them to have larger, more powerful instruction sets (some, for instance, can do multiplication directly, which the 8-bit processors can't), to handle larger numbers, and to run some programs (especially those involving large-number arithmetic) more quickly. But they cost more—and need more memory.

There's also a chip called the Pascal Microengine, which is designed to work specifically with PASCAL, a computer language of growing popularity. So far, there are few application programs available in PASCAL for microcomputers, but you can expect to see more of them in the next few years.

Structure and Expandability. Any computer system worth its salt is designed to allow expansion. Your needs may grow or change; your budget will certainly grow, allowing you to make additions piecemeal.

Computer systems can be expanded in a variety of ways, and a given computer may use several of them. The simplest way to expand a system is to plug more integrated circuits into sockets already provided for them. This is usually done to expand RAM and ROM memory, and only for moderate expansions. Many single-board computers use this method, but so do some larger ones.

A more popular and more versatile route to system expansion is to plug in additional circuit boards. This implies that the computer will have some sort of *bus structure*, which is a group of signal, data, address and power lines into which boards can be plugged in any order. Several bus systems are in use, some are used in just one model of computer, other (such as the S-100) are used in many.

Boards are available for a very wide variety of purposes: to expand memory; to add more I/O circuits for additional terminals, printers and the like; to generate speech or sounds; to accept voice input; to tell the computer what time it is; to allow the user to build circuits of his own; to control other devices; to communicate by phone with other computers and terminals; to test integrated circuits; to add graphics capabilities; to send and receive Morse code; to interface with computers using other buses; to speed math processing; and many more.

Some computers, chiefly very compact ones, require a separate "box" to hold more than a minimum of extra memory, I/O and other circuits. Others combine approaches, with an expansion box built around an S-100 bus.

Peripherals. Much system expansion occurs outside the computer. With the right programs and I/O circuits, a computer (even the kind whose built-in keyboard and video screen make it a terminal unto itself) can support several terminals around an office or house. For a very few machines, there are programs available which allow several terminals to operate at once.

Even a one-terminal system can frequently use an add-on printer, for all the reasons already cited. But the application has a lot to do with which printer should be selected. The main choices are between dot-matrix and character printers; between impact, electrosensitive and thermal printing systems; and between printers offering upper-case (capitals) only and those offering both upper and lower case.

To record computer programs, type mailing labels, do shopping lists and invoices, among other things, an upper-case printer is quite adequate. But for typing letters and manuscripts, both upper- and lower-case are needed. Upper and lower case are also recommended for any long text—pure upper-case text is harder to read. Lower case costs more, but the readability is worth it.

Printers that make fully-formed characters, like a typewriter, also produce more readable, more personal-looking text than those which form letters from a matrix of separate, unconnected dots. Dot-matrix printers tend to be cheaper and faster, though, so you may prefer them for short texts and program dumps. The more dots, the more readable: a 5X7 dot matrix is rather crude to read; a 9X12 matrix almost as easy as formed characters, though it still has a computer-printed look.

Impact printers work on the same principle as the typewriter, pressing inked ribbon against the paper. Th makes them the noisiest of printers, t allows them to use any ordinary pap (your letterhead, for example) and produce carbon copies.

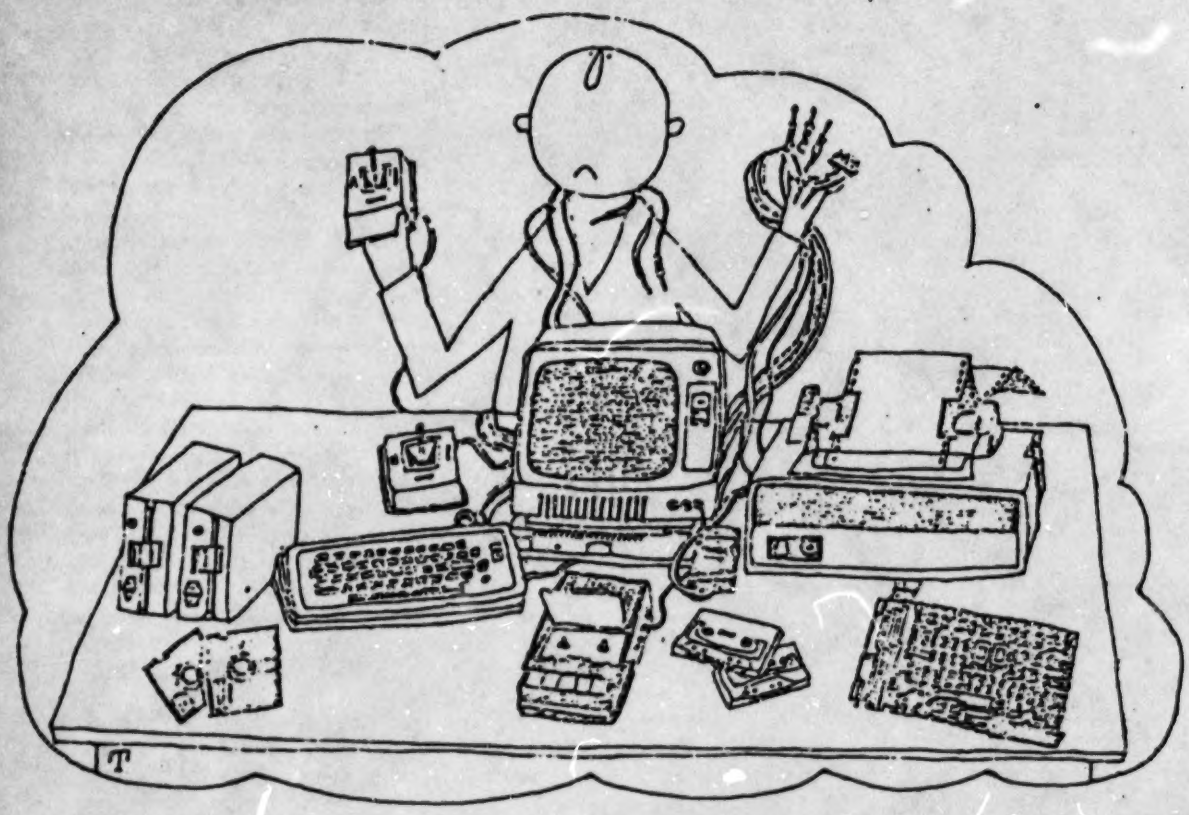
Thermal and electrosensitive printe by contrast, are quiet, require no ribb replacement, make only one copy a time, but require special paper that's ten available only from computer-sup dealers. That means they cost more the long run than plain-paper impa printers, though electrosensitive pr ers are about the least expensive or to buy. However, electrosensitive pa; is silver-colored, not white, wh makes it hard to read. Both thermal a electrosensitive printers are only av able as dot-matrix types.

Printers can also be used for gra ics. At the simplest level, this mea mapping your printout as you wo when "drawing" pictures with a ty writer. "Daisy-wheel" character pr ers, such as the Qume and Diablo, c be used to draw even finer pictur since their printheads can be advanc by 1/10 or 1/12 the width of a norm printed character, but the use of s ware to do this with microcomputers not common yet. Many dot-matrix print can be used to draw dot pictures, to

Modems are another useful acces ry, allowing your computer to comm cate with others by telephone. Or nate-only modems, the least expens type, let your computer call up othe Originate/answer types also let oth call you up. Some of the latter type a have "auto answer" facilities, so t can answer calls even in your absenc

The Systems Approach. When buy a computer, you're not just buyin computer. You're starting a *system*. your choice should be governed by entire system it belongs to, and h well that system suits your applicat Can the system be expanded to k pace with your future needs? Can get the peripherals you need—d drives, modems, printers, device con boards, or whatever? How easily you add any extra memory you n need, and at what cost? How m companies supply equipment to with this system? And, most impor of all, is software available to make system do what you want and need do? If the answer to all these quest is yes, then you've found the right tem. This is an area where the help good dealer is well worth seeking.

68



ETTING YOUR SYSTEM UP AND RUNNING

BY IVAN BERGER

ETTING a computer system up and running is a process that begins ore you buy it: you must make sure t all the pieces you need are included he system and that they'll all work in mony. Even when they don't, there's ly cause for panic. The problem usu- is not one of product defects, but of ting those products properly with one ther. In a simple system, such dif- ficulties may never arise at all.

or some computers start-up is simply natter of plugging it in, turning it on, following instructions in the manual stablish a dialog with it. This is the with a small, self-contained system combines a terminal keyboard and o display in one package. For many rs, it's almost as simple: the monitor en and cassette recorder are in sep- housings that must be plugged computer itself.

t for some others, you'll need to do , and possibly even buy additional ponents. Some popular computers, ample, don't include video monitor ns. You'll have to purchase one computer dealer will be sure to

have some), or have a home TV set modified by a serviceman to include a direct video input, or buy an *r-f* kit to convert the computer's video output into a TV-channel signal that your television receiver can pick up. Some computers have modulators built in, for use with regular TV sets. And if you have a video cassette recorder, you may be able to use its video input and built-in modulator to put computer images on your TV set's screen—or to record them. This does not work with all computer/VCR combinations, but it does work well with some.

Displaying your computer's output on a standard TV set's screen costs far less than buying a separate monitor. But using a monitor has advantages, too: it leaves the TV screen free for others in the family to use. Moreover, it will likely give you a sharper image because the signal bypasses the modulator and the set's tuning section and, in many cases, because the monitor has higher bandwidth than an ordinary TV set.

Some monitors are just TV sets with the tuners removed (yet with a higher price, alas). Others are specifically designed for higher resolution than TV receivers have. The most commonly cited monitor specification that relates to resolution is bandwidth. A bandwidth of 4.5 MHz is probably the rock-bottom minimum to look for in such a monitor, while anything over 6.6 MHz will be wasted in most home-computer applications. (Computers with high-resolution or high-density graphics output may require more than 6.6 MHz, though; consult your system's manual to be sure.)

Separate Terminal. Many computers come with neither keyboard nor video output. You can communicate with them in either of two ways: by equipping the computer with an input port for a separate keyboard and a video output-board to feed a monitor screen or *r-f* modulator, or by equipping it with an input/output port through which it can communicate with a terminal.

A terminal is simply a keyboard combined with a video display or printer, with provision to display or print both your keyboard input and the computer's replies. Computers which do not have

their own keyboards and screens or video output will either come with the necessary ports or have them available as accessories.

Accessory video and input/output (I/O) boards (as well as accessory boards for memory and other purposes) may either be made for specific computers or be designed to plug into most computers using a given "bus." A bus is a standardized layout of signal lines and connectors that allows circuit boards of many kinds to be added and subtracted at will. The S-100 bus is the most common one among small computers. When adding boards, it is important to check whether the total power drawn by all boards in the system will still be within the power supply's capacity.

Serial and Parallel Input. There are two types of input/output circuit: serial and parallel. Most keyboards require parallel connection, with all eight data bits per data "word" (a character or command) reaching the computer simultaneously, each through its own wire. This is the efficient and inexpensive way, since the computer also processes data in parallel internally.

Most computers use serial connection, with the bits sent one at a time over the same pair of wires. Serial I/O circuits cost more, but are easier to use for long wiring runs. Computer terminals are usually serial devices. Printers are often available in both serial- and parallel-connection models. The extra-cost serial option is for use when the computer and printer must be some distance apart.

Once you've set up whatever devices you need to communicate with your computer, what you "say" depends less on the system hardware than on the software, or programs, which tell it what to do and how to respond to your input.

Most computers on the market today have a high-level language (usually BASIC) in read-only memory (ROM). In some, the computer is ready to talk BASIC with you as soon as you turn it on. In others, you need a key-stroke or two to enter BASIC.

With some computers, though, you have to load BASIC into your main memory from a cassette recorder, floppy disk unit, or paper tape (though paper tape is rare, today). Such computers will usually have a different program in ROM and a monitor program (not to be confused with a TV monitor), which tells the computer how to load new programs from tape or disk and how to interpret your keyboard commands. A very few computers, however, require that you enter a "bootstrap" program into the computer in order to load the monitor from tape or disk. Such computers must have front panels allowing direct input, either in octal or hexadecimal numbers from a keypad, or in binary numbers from a bank of switches (one switch per bit). Even front-panel machines nowadays will usually have ROM monitors, rendering the bootstrap unnecessary. But the front-panel is useful in de-bugging programs and hardware.

Floppies and Cassettes. Loading a program in from tape or disk requires a cassette or floppy-disk system to load them from. Many computers today have one or the other built in. Most, though, require that they be added externally.

Of these, many come with a separate recorder, while others include only an interface for storing data and programs on a standard audio cassette recorder.

Adding floppy-disk systems to your computer (or adding cassette, if your computer lacks it) is usually a matter of inserting a special board into the computer, then connecting the disk or cassette unit to it. (The new, hard disk systems now appearing also require such boards.) As with I/O boards, these may be designed for specific computers or for common busses—but there are also a few which connect to regular I/O ports.

Having a cassette or floppy-disk system does not necessarily mean that you can readily swap programs or data with other floppy- or cassette-equipped hobbyists. There are a wide variety of systems and "standards." Only computers with both the same type of processor (e.g., 6502, 8080/Z80, 6800, 1802) and floppy or cassette systems operating on the same standards can interchange programs in this way.

There are many variations to watch out for. Different cassette systems may record different sets of tones on the tape, may transfer data at different speeds, and may use different combinations of characters for the start and end of each taped record. Record and playback volume settings may be critical when interchanging tapes from systems which are theoretically identical.

With floppy disks, there are even more variations to note. Floppies come in both full (8-inch) and mini (5¼-inch) sizes, and some mini-floppies record more tracks on the disk than others. Some systems use "hard-sectored" disks, with a ring of small holes surrounding the large, central one; oth are "soft sectored," with but a singl dex hole. Several systems are avail in both single- and double density sions, with single-density unable to r double-density disks.

Despite this wide range of varia there is far more standardization in py systems than in cassettes. Use S-100 bus systems are fortunate in respect: the North Star and Micro mini-floppy systems, and 8-inch fl systems using the CP/M operating tem have become *de facto* stand with programs available in these for from many vendors. The CP/M disk erating system (DOS) is even avail for North Star and Micropolis disk ups. Neither, though, can read written by the other, or interchange with full-size CP/M systems.

There are also several S-100 bus sette "standards," such as Tarbell, sas City (the only one officially ad as a standard, and one of the used) and CUTS. But none of the as popular now as they were before systems became common.

Computers of vast popularity, su the TRS-80, Apple, Pet, KIM-1 and ceror become standards unto t selves, of course, with software able for them from many sources. T are even crossbreeds: S-100 bus faces are available for all of these puters (though not all S-100 board be used with some of them). And s al companies have appropriated for the TRS-80.

Interfacing. Computer boards ripherals and other extras are not li fi components: You can't just plug together and automatically expect to run. Usually, a few adjustment needed for smooth operation.

Those adjustments may simp volve flicking a few switches, or m jumper wires from one hole on a to another. Or they may involve modifications to programs or hard But they always require good doc tation—operating and service ma for your hardware, source listing other detailed manuals) for softw so you can figure out just what to d

It also pays to draw on outsid and advice, when available. A con club, if there's one near you, will of potential helpers. Your dealer o ers will also help. The more yo from one dealer, of course, the helpful he'll be. That's not just be he's made more money from

though that is a factor). The more of the system that came from him, the more ly he is to be familiar with whatever ponents aren't working with each other properly.

Problems are easier to deal with when you know what to expect. So here's a list of the major areas likely to need attention when setting up a system:

Memory. Every machine-language program (including monitor programs, interpreters or compilers for BASIC and other languages, assemblers, disk-operating systems, and so on) will be designed to reside and run in a specific block of memory addresses. Make sure when you buy your software that programs which will be in your system's memory at the same time use different blocks of memory. This would include both programs designed to work together (editors and assemblers for example) and all programs in ROM or PROM.

Be certain, too, that your memory boards are set to the addresses that your programs require, and that no two memory boards are set to the same or overlapping addresses. Changing a memory board's address is usually a matter of moving a jumper wire or resetting some small switches.

Using a fast processor (such as a 4-MHz Z80) with slow memory can cause problems, too. Many memory boards provide for "wait states" to slow down the program long enough for the memory to catch up with the processor.

I/O. Input/output ports have addresses, too. These addresses must also agree with the programs that use them. Setting up your I/O boards' addresses to match your software is as easy as changing a memory board's address. But changing your software's I/O address calls can be easy, too. (Changing a program's memory addresses is often a major undertaking.) So which should you change if the hardware and software disagree?

Where change is called for, the answer may be to split it between software and hardware. Programs in ROM or PROM ("firmware") can't be modified readily, so you'll have to set your port hardware addresses to match those programs. Then modify any software which calls for different port addresses, so it matches the way you've set your hardware up.

Serial settings. Serial I/O may need some further tinkering. First, the baud rate (data transmission speed) of both the port and the peripheral connected to it must be matched. Other options must also be made to agree between the port and the peripheral. Some systems are "full duplex," with the terminal not showing your keyboard input until it's reached the computer and been echoed back; others are "half duplex," printing or displaying the keyboard input as soon as it's entered. Some serial circuits use RS-232 signal levels and connections, while others use 20-mA current-loop levels and connections instead. Parity bits (transmitted as a check against transmission errors) and stop bits (marking transitions between data words) must be set to match as well.

If one component offers no choice in any of these matters, then all other components should be set to match it. Otherwise, the choices can be made fairly arbitrarily, so long as the same choice is made for any two devices which communicate with one another.

Timing requirements. Microcomputer components must work in synchronization with each other. Since they also run at lightning speeds, that makes their timing as difficult as it is critical. The more different sources your boards came from, the more likely such problems are to occur. Getting the timing right can require trial-and-error replacement of capacitors or resistors in timing circuits—strictly a job for the knowledgeable.

Bus variations. Scores of manufacturers made module boards for the S-100 bus for several years before the IEEE defined a standard configuration and standard signal formats for it. As a result, boards from different manufacturers may use the same bus lines for different signals, or one manufacturer's board may require a signal that another manufacturer's board puts out. (This is especially true of Z80 CPU boards, many of which do not generate all the signals which 8080 boards do.) When contacting the manufacturer of any board which seems to be malfunctioning, be sure to list all the other boards in your system so that he can spot troubles which lie in a conflict between boards.

Software. A computer is useless without programs. You'll doubtless buy a monitor program, a high-level language interpreter or compiler (probably BASIC) or both when you get your machine. But you'll quickly find that you need others to perform whatever tasks you purchased your computer for. You can write these programs yourself, key them in from printed books or listings, or buy them on tape or disk.

Writing your own programs is both the most time-consuming and most satisfying of the three. And once you develop the necessary programming skills (often one of the main reasons for buying a computer in the first place), you'll wind up with programs that are custom-designed for your particular needs and ways of thinking.

Keying in a program from a printed listing may do nothing for your ego, but it's faster. You may also find that programs written for other computers may require some modifications to run on yours. There are, for instance, more dialects of BASIC than there are of Chinese, and programs written in one dialect may use commands inscrutable to other dialects. (David Lien's *The BASIC Handbook* is an excellent guide around such problems.)

Entering a program from a tape or disk is fastest and easiest. But to work, the program must be distributed in the disk or tape format your system uses. If written in machine language, it must be configured for your memory and I/O addresses (though you can load it first, then alter it). And if in BASIC or some other high-level language, it must be written for your specific language interpreter or compiler. This can be more critical than it looks. Programs saved on tape from 8K Altair BASIC, for example, may not run properly in 12K Altair BASIC. (The 8K program would work perfectly in 12K, though, if it were entered at the keyboard from a printed listing, not loaded from the tape.)

When you buy a program, you should try it in the store, first, to make sure it does everything you want in a way that you find natural and convenient. If possible, try it on a system exactly like yours. Since most dealers sell programs for the systems that they sell, you'll probably be able to try them at your dealer's on a system similar to yours. The main exception to this is the Radio Shack TRS-80. Only Radio Shack stores sell it, and they sell only Radio Shack programs. But the TRS-80's popularity has led many mail-order suppliers to sell programs for it, though they do not naturally have the "walk-in" facilities or local location to permit a demonstration.

In conclusion, don't think that, just because so many finicky details may need attention, a computer system is a chronic invalid. Think of it rather as much like a baby, requiring fussing and attention before it can do anything appreciable on its own. And some computers—like some babies—give you pure pleasure, with no trouble at all. ○

COMPUTER GLOSSARY

ACCUMULATOR—In a microprocessor, the internal register in which logical operations are performed and the results initially stored; characters may also be input to or output from the accumulator.

A-D—Conversion of continuous, analog data (like meter readings) into digital form that computers can read.

ADDRESS—The number used to refer to a specific byte in memory or to an input or output port.

ALPHANUMERICS—Computer output or input in the form of letters and numbers rather than graphs or drawings.

ANALOG—Originally, the physical representation of numerical quantities in terms of motion, voltage, resistance, etc. By extension, any data which changes in a smoothly varying way, rather than changing in discrete steps as digital data does.

ASCII—American Standard Code for Information Interchange, a seven-bit code used by most microcomputer equipment to represent alphanumeric characters.

ASSEMBLER—A program that converts assembly-language into machine language.

ASSEMBLY-LANGUAGE—A computer language that uses easily remembered groups of letters as commands instead of the "ones" and "zeros" a computer understands, e.g., JNZ (jump if not zero) instead of 11000010.

BASIC—Beginners All-purpose Symbolic Instruction Code, the most common high-level language used on home computers.

BAUD RATE—The number of signal elements transmitted per second. With the transmission systems used in microcomputers, equal to bits per second (bps).

BINARY—The number system with base 2. There are only two digits in the system: "0" and "1".

BIT—A binary digit. The smallest bit of information possible.

BUS—A group of wires connecting CPU, memory and I/O for exchange of information.

BUS STRUCTURE—A fixed arrangement of the wires of a bus.

BYTE—A computer word eight bits long; it has 2^8 (256) possible values. Most home computers use one-byte instructions, and a data bus one byte wide.

COBOL—Common Business-Oriented Language, a computer language designed for business programming.

COMPILER—A program that converts programs written in high-level languages, like BASIC, into a program that a computer can run directly or with the aid of a shorter "run-time" program.

CPU—Central Processing Unit, the circuit or subsystem which actually does the computing.

CRT—Cathode Ray Tube, a TV-type screen which may be used by a computer to display its alphanumeric and graphic output.

D-A—Conversion of digital data to be continuous, analog form; a circuit to perform this conversion.

DATA—The information in a computer program that the computer processes, as opposed to the information that tells the computer what process must be done.

DISPLAY—A device that shows the computer output or status visually instead of on paper. The most frequently used displays are CRTs, or multi-segment LEDs (like calculators).

EPROM—Erasable PROM, a type of PROM which can be erased with ultraviolet light and then re-programmed.

FIRMWARE—Software (programs) stored in ROM or PROM memory.

FLOPPY/FLOPPY DISK/FLOPPY DISKETTE—A thin disc of magnetic material like recording tape that is used for recording and storing computer programs and data.

FORTRAN—FORmat TRANslator, a high-level language designed for scientific programming.

GRAPHICS—The processing, input and output of data other than alphanumerics or control or status functions (for example, pictures, images and graphs).

HARD COPY—Computer output that is printed rather than output on a display.

HARDWARE—The computer equip
self, as opposed to its programs (sof

HEX/HEXADECIMAL—A number
with base 16. The sixteen digits are
4, 5, 6, 7, 8, 9, 0, A, B, C, D, E, and F.

HIGH-LEVEL LANGUAGE—A compu
guage that humans can understan
and that a computer can translate
machine-language form it understand
or more steps. Many instructions in
level language require that the compu
form more than one computer op
Some of the more common high-le
guages are: BASIC, COBOL, FORTR
PASCAL.

INPUT—The information that is fed
computer; it may contain data, instr
or both.

INSTRUCTION—The portion of the i
tion fed into a computer that tells the c
er what to do with the other info
(data) it receives.

INTERFACE—A circuit to form the
connection between a computer and
other device.

INTERPRETER—A program which
decodes and executes a high-leve
gram. Unlike Compilers, interpreter
be loaded into a computer both wh
user program is being entered and v
is run.

I/O—Input/output. A) The equipmen
to put information into or take informat
of a computer. B) The information itse
is given to or taken from a computer.

KEYBOARD—A group of keys. Among
who work with computers, it is usuall
to mean a typewriter-like layout of ke
numbers, letters, punctuation and othe
bols) plus the accompanying electr
that is used to input information into a
puter. Other arrangements of keys are
ly referred to as keypads.

"K"/KILOBYTE—2^{10} (1024) bytes. M
is usually reckoned in kilobytes. "K" s
for "binary thousand" (1024), while o
decimal thousands are represented by
case "k".

MACHINE CODE/MACHINE LANGUA
instructions that are in binary form and
ally understood by a computer without t

ecoding. These instructions are usually in he form of 8 bit (one byte) words in home omputers, however some instructions, in- rating data or addresses, may be two e bytes long.

ASS STORAGE—Recording systems for olding or storing programs or data not re- uired for immediate use. Such information ust be read into the computer before it can e used. Common forms of mass storage sed with home computers are cassettes nd floppy disks.

EMORY—The part of a computer dedicat- d to storing programs and data. Memory is rganized as words (usually 8 bits) each of hich has a unique address so that the com- uter may select any word it needs by using he address of that word.

INI FLOPPY—A 5¼-inch floppy disk. tandard floppy disks are 8 inches in diame- r.

ODEM—An I/O device that permits the omputer to receive or transmit information ver telephone lines.

ONITOR—A) A CRT screen and associated lectronics which may be used for computer isplay. B) A program that instructs the com- uter how to do "housekeeping" tasks such s: handling input or output; changing, storing writing), or reading the contents of memory; ic.

VOLATILE—Memory that retains its ontents even if no power is supplied to it. See RAM.)

CTAL—A number system with base 8. The ight digits are: 0, 1, 2, 3, 4, 5, 6, and 7.

UTPUT—The information sent out by a omputer. It may be visual (printed or dis- layed), aural (sound or music), electrical control for a motor), etc.—any communica- on from a computer to the outside world.

APER TAPE—A mass storage system sing paper tape with eight hole-positions epresenting the eight bits of each byte tored.

ARALLEL—A type of I/O in which each bit f a computer word is transmitted over a sep- rate wire simultaneously. (See serial.) A omputer's internal information flow is also ormally in parallel form.

ASCAL—A comparatively new computer anguage, now becoming available for ome computers.

ERIPHERALS—The parts of a computer ystem outside of the computer proper, such s: terminals, displays, printers, etc.

ORT—The I/O circuit that connects a com- with a peripheral. Each port has an ad- ss (number) by which the computer can stinguish it from other ports. Ports may be erial or parallel.

RINTER—A mechanism that prints the out- ut of a computer. Printers with keyboards re called printing terminals.

PROCESSOR—A set of circuits capable of performing the essential functions of a computer CPU. In the case of a microprocessor (MPU), these circuits are combined into one or a few integrated circuits.

PROM—Programmable ROM, non-volatile memory which can be programmed by a user. Two subdivisions of this class are EPROM (Erasable PROM) and EAROM (Electrically Alterable ROM) which are erasable with ultraviolet light or electrical impulses respectively. EPROM and EAROM may be re-programmed after they are erased.

RAM—Random Access Memory, any type of memory that may be written into or read from. It is randomly accessible if it takes the same quantity of time to reach any address, independent of location. In contrast, a Serial Access Memory is one in which the time required to get to one address from another is dependent on how far the addresses are from each other. Information stored on cassette is serial, for example. Most RAM is volatile in that the contents rearrange themselves randomly when the power is shut off.

R-F MODULATOR—Radio-Frequency Modulator, a device that converts video information into frequencies acceptable to a TV set so that the TV may be used as a substitute for a video monitor.

ROM—Read Only Memory, memory that can only be read from, not written into, because its contents have been fixed during manufacture. The term is often loosely applied to any non-volatile memory, especially members of the PROM (user programmable) family.

SERIAL—A type of I/O port in which all data is transferred over a single pair of wires, one bit at a time. In practice, the data words are sent in a specific pre-selected format to make each word distinguishable from the others that precede and follow it.

SOFTWARE—Computer programs, sets of instructions that tell the computer what to do and how it is to be done.

SYSTEM—The set of electronics and machinery that is assembled for the computer to perform its tasks. It includes not only the CPU or MPU, but the memory, I/O and peripherals.

TERMINAL—A peripheral device combining a keyboard for human input to the computer with a display or printer for the computer's output.

WORD—A computer word, the largest number of binary digits that a computer can handle simultaneously. Most home computers can handle a maximum of eight bits (one byte) of data at the same time. Some newer microcomputers can handle words of up to 16 bits.

WORD PROCESSOR—A system which accepts words and text as input, and which allows that text to be modified, merged with other texts, and output in a desired format.

". . .so I said, SZE and PR then GOTO or at least GOSUB . . ."

Directory of Microcomputer Products Manufacturers (u.s.!)

ALF PRODUCTS, INC.
1448 Estes, Denver, CO 80215

ALLIED COMPUTERS
46th St., 9th Ave., 11, Ashok Nagar, Madras-600 083, India

ALPHA MICRO
17881 Sky Park North, Irvine, CA 92714

APF ELECTRONICS, INC.
444 Madison Ave., New York, NY 10022

APPLE COMPUTER INC.
10260 Bandley Dr., Cupertino, CA 95014

ATARI, INC.
1265 Borregas Ave., PO Box 9027, Sunnyvale, CA 94086

ATV RESEARCH
13th and Broadway, Dakota City, NE 68731

AUM-IDEAS
P.O. Box 2582, Richardson, TX 75080

AUTOMATED INDUSTRIAL MEASUREMENTS, INC.
P.O. Box 125, Wayland, MA 01778

CALIFORNIA COMPUTER SYSTEMS
309 Laurelwood Road, Unit # 18, Santa Clara, CA 95050

CENTRAL DATA CORPORATION
P.O. Box 2530, Station A, Champaign, IL 61820

CENTRONICS DATA COMPUTER CORPORATION
Hudson, NH 03051

CGRS MICROTECH
P.O. Box 368, Southampton, PA 18966

CHRISLIN INDUSTRIES, INC.,
31352 Via Colinas, #102, Westlake Village, CA 91361

COMMODORE BUSINESS MACHINES LTD.
901 California Ave., Palo Alto, CA 94304

COMPRINT, Computer Printers International, Inc.
340 East Middlefield Rd., Mt. View, CA 94043

COMPUCOLOR CORPORATION
5605 Peachtree Corners East, P.O. Box 569, Norcross, GA 30091

COMPUTALKER CONSULTANTS
P.O. Box 1951, Santa Monica, CA 90406

CROMEMCO, INC.
280 Bernardo Ave., Mountain View, CA 94040

D.C. HAYES ASSOCIATES, INC.
10 Perimeter Park Dr., Atlanta, GA 30341

DELTA PRODUCTS, INC.
1953 East 28th St., Signal Hill, CA 90806

DIGITAL SPORT SYSTEMS
Seventh and Elm Sts., West Liberty, IA 52776

DYMA ENGINEERING
Box 1897, Taos, NM 87571

ELECTRONIC CONTROL TECHNOLOGY
763 Ramsey Ave., Hillside, NJ 07205

ELECTRONIC PRODUCT ASSOCIATES, INC.
1157 Vega St., San Diego, CA 92110

ELECTRONIC SYSTEMS
P.O. Box 21638, San Jose, CA 95151

EVENTIDE CLOCKWORKS, INC.
265 West 54th St., New York, NY 10019

EXIDY, INC.
2399 Garcia Ave., Mountain View, CA 94043

F&D ASSOCIATES
1210 Todd Rd., New Plymouth, OH 45654

GIMIX, INC.
1337 West 37th Place, Chicago, IL 60609

(BILL) GODBOUT ELECTRONICS
Box 2355, Oakland Airport, CA 94614

HAZELTINE CORPORATION
Computer Terminal Equipment, Greenlawn, NY 11740

HEATH/SCHLUMBERGER DATA SYSTEMS
Hilltop Rd., St. Joseph, MI 49085

HUH ELECTRONICS
1429 Maple St., San Mateo, CA 94402

IMAGE-21
1303 Broadway, Dakota City, NE 68731

INNOTRONICS CORPORATION
Brooks, Rd., Lincoln, MA 01773

INNOVATIVE TECHNOLOGY
510 Oxford Park, Garland, TX 75043

INTERACT ELECTRONICS, INC.
P.O. Box 8140, Ann Arbor, MI 48107

INTERTEC DATA SYSTEMS
2300 Broad River Rd., Columbia, SC 29210

ITHACA AUDIO
P.O. Box 91, Ithaca, NY 14850

JADE COMPUTER PRODUCTS
4901 West Rosecrans, Hawthorne, CA 90250

JHM MARKETING ASSOCIATES
4340 Campus Dr., Suite 212, Newport Beach, CA 92660

LARKS ELECTRONICS & DATA
P.O. Box 22, Skokie, IL 60077

LEAR SIEGLER, INC.
714 North Brookhurst St., Anaheim, CA 92803

MATROX ELECTRONICS, LTD.
5800 Andover Ave., Montreal, Quebec H4T 1H4, Canada

MECA
P.O. Box 696, 7026 Old Woman's Spring Rd., Yucca Valley, CA 92284

MICRODASYS
P.O. Box 36051, Los Angeles, CA 90036

MICRO PERIPHERALS, INC.
2099 West 2200 South, Salt Lake City, UT 84119

MICROPOLIS
7959 Deering Ave., Canoga Park, CA 91304

MICROPRODUCTS
2107 Artesia Blvd, Redondo Beach, CA 90278

THE MICRO WORKS
P.O. Box 1110, Del Mar, CA 92014

MK ENTERPRISES
8911 Norwick Rd., Richmond, VA 23229

MOUNTAIN HARDWARE, INC.
300 Harvey West Blvd. Santa Cruz, CA 95060

MULLEN COMPUTER PRODUCTS INC.
Box 6214, Hayward, CA 94545

NATIONAL INSTRUMENTS
9513 Burnet Rd., Austin, TX 78758

NATIONAL MULTIPLEX CORPORATION
3474 Rand Ave., Box 288, South Plainfield, NJ 07080

NESTAR SYSTEMS, Inc.
430 Sherman Ave., Palo Alto, CA 94306

NETRONICS RESEARCH & DEVELOPMENT LTD.
333 Litchfield Rd., New Milford, CT 06776

N.N.C. ELECTRONICS
15631 Computer Lane, Huntington Beach, CA 92649

NORTH STAR COMPUTERS
2547 Ninth St., Berkeley, CA 94710

OAE, Oliver Advanced Engineering, Inc.
676 West Wilson Ave., Glendale, CA 91203

OBJECTIVE DESIGN, INC.
P.O. Box 20325, Tallahassee, FL 32304

OHIO SCIENTIFIC
1333 Chillicothe Rd., Aurora, OH 44202

OPTIMAL TECHNOLOGY, INC.
Blue Wood 127, Earlysville, VA 22936

OTTO ELECTRONICS
P.O. Box 3066, Princeton, NJ 08540

PARATRONICS, INC.
800 Charcot Ave., San Jose, CA 95131

PERCOM DATA COMPANY, INC.
211 North Kirby, Garland, TX 75042

PICKLES & TROUT
P.O. Box 1206, Goleta, CA 93017

RADIO SHACK, Div. of Tandy Corporation
1400 One Tandy Center, Fort Worth, TX 76102

RCA Solid State MARKETING
New Holland Ave., Lancaster, PA 17604

ROCKWELL INTERNATIONAL, Microelectronic Devices
P.O. Box 3669, Anaheim, CA 92803

SDS TECHNICAL DEVICES LTD.
P.O. Box 1998, Winnipeg, Canada R3C 3R3

SD SYSTEMS
P.O. Box 28810, Dallas, TX 75228

SILVER SPUR
13552 Central Ave., Chino, CA 91710

SMOKE SIGNAL BROADCASTING
31336 Via Colinas, Westlake Village, CA 91361

SOLID STATE MUSIC
2116 Walsh Ave., Santa Clara, CA 95050

SOROC TECH, INC.
165 Freedom Ave., Anaheim, CA 92801

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 West Rhapsody, San Antonio, TX 78216

SPACE-TIME PRODUCTIONS
2053 North Sheffield, Chicago, IL 60614

SZERLIP ENTERPRISES
1414 West 259th St., Harbor City, CA 90710

TARBELL ELECTRONICS
950 Dovlen Place, Suite B, Carson, CA 90746

TECHNICO, INC.
9130 Red Branch Rd., Columbia, MD 21045

TELESENSORY SYSTEMS, INC.
1889 Page Mill Rd., Palo Alto, CA 94304

TELETEK
9767F Business Park Dr., Sacramento, CA 95827

TELETYPE CORPORATION
555 Touhy Avenue, Skokie, IL 60077

TERMINAL DATA CORPORATION OF MARYLAND
11878 Coakley Circle, Rockville, MD 20852

TEXAS INSTRUMENTS INCORPORATED
P.O. Box 53, Lubbock, TX 79408

THINKER TOYS
5221 Central Ave., Richmond, CA 94804

TRINEX DESIGN
1201 Marlkress Rd., Cherry Hill, NJ 08034

ULTRA-VIOLET PRODUCTS, INC.
5100 Walnut Grove Ave., San Gabriel, CA 91778

U.S. ROBOTICS, INC.
1035 West Lake St., Chicago, IL 60607

VECTOR ELECTRONIC COMPANY, INC.
12460 Gladstone Ave., Sylmar, CA 91342

VECTOR GRAPHIC, INC.
31364 Via Colinas, Westlake Village, CA 91361

VOLKER-CRAIG LIMITED
266 Marsland Dr., Waterloo, Ontario N2J 3Z1, Canada

WAMECO, INC.
111 Glenn Way, #8, Belmont, CA 94002

WESCORP, Div. of DAL Industries, Inc.
1155 Terra Bella Ave., Mt. View, CA 94043

ELECTRONIC EXPERIMENTER'S HANDBOOK
1980

ADVERTISERS INDEX

READER SERVICE NO.	ADVERTISER	PAGE NUMBER
1	Byte	103
	Cleveland Institute of Electronics, Inc.	35,36,37
2	Continental Specialties	2
3	Creative Computing Magazine	101
4	Datak Corporation	113
8	Electronics Book Club	5
6	Exidy, Inc.	8,9
7	G C Electronics	4
5	Heath Company	53,54,55
9	International Crystal	109
10	Jensen Tools & Alloys	87
	National Radio Institute	17,18,19
11	Ohio Scientific Instrument	Cover 4
12	OK Machine & Tool Corporation	81
13	Percom Data, Inc.	Cover 3
14	RCA Solid State	1
15	Trio-Kenwood Communications Inc.	Cover 2

E

"The following programs were used during our project. Most of these programs were used by the students during or before school, lunch and after school. We decided that this was the best time for them as many consisted of games."

1. Blackjack, Radio Shack

This is a card game which is useful for learning betting and how to play blackjack. The student interest was about 3 on a scale of 1 to 5.

2. Aerodynamics, Softside.

This program teaches the basic parts of any airplane.. It is one of the few CAI programms we have but was not used much as it takes too long to load. Student interest 3.

3. Stimulating Simulations, Creative Computing

On this tape there are ten simple games. Two of them are similar to Hurkle and are therefore useful for mapping skills. Student interest 3.

4. Bomber, Softside

This game is manipulative in nature and therefore has a high student interest (4). It is similar to Brick out.

5. Taipan,

This simulation is a trading game. The object is to be such a good business man that you earn a million dollars in the opium trade (we changed our version to jade before the students saw it). Student interest is 4 (but loading this game takes a long time)

6. Lemonade Stand, softside.

This simulation is a buying and selling game. Student interest 3.

Programs (continued - page 2)

7. Horse Corral, Softside

The object of this game is to set up fences to enclose a randomly moving horse with as little a fence and time as possible. Student interest 2.

8. Concentration, C load.

This is the old card game, modified to reveal parts of numbers underneath the cards. The student must then guess the five digit number to win prizes. Student interest 5.

9. Grand Prix, Softside

Car race game under arrow control. It is very difficult to win, therefore, student interest low 2.

10. Play it Again Sam (Simon) CLOAD

Follow the sequence game - Needs audio hook up which was under repairs for part of the time. Student interest 3.

11. Syzygy (blocade) C load

High student interest 4.

12. Missile,

Boat shoots at plane, plane shoots at boat. Student interest 2 (graphics move slowly)

13. MicroMusic, Radio Shack

Play music by writing letters and numbers for notes. Student interest high for making random tones but few wished to read the manual to see all of the possibilities perhaps because of the complicated methods of writing real music offered by the program.

14. Animal, 101 Basic Computer games.
This is an interesting game because the computer "learns" information from what the student inputs. The program needs Disk to be fully effective as an educational tool. Student interest 2.

15. Lifeboat 80 U.S.
In this simulator the student is in charge of a lifeboat and must decide life and death situations. Student interest 3.

16. Star Pilot Radio Shack
Manipulative game, Student interest 4.

17. Quick Watson, Radio Shack
Good game of deductive logic. Student interest 2.

18. Code breaker (similar to Mastermind), local programmer)

19. Eliza Radio Shack
The computer seems to ask intelligent questions.
Student interest 3.

20. Maze Chase Cload
Manipulative game. Student interest 3

21 Arithmetic program, local programmer
Adds, subtracts, multiplies, divides on a number of difficulty levels. Student interest 2 (because ma. of manipulative games like this)

22. Dragrace, Softside.
Manipulative game simulating shifting gears using a tachometer as a guide. Student interest 4 (boys)

23. Westward 1847 Softside
A very difficult simulation to win which involves trying to stay alive by getting your own food. Student interest 2.

24. Defend, Cload
A manipulative game similar to space invaders but lacking the sophistication. Student interest 3.

25. Saucer Radio Shack
Manipulative; shoot at the saucer. Student interest 3.

26. Collision, Softside
The game which students showed the most interest in because of its manipulative nature. They tried for higher and higher point values. I was even asked to make it harder by increasing the speed and traffic density. Student interest 5.

27. Baseball, Softside.
Two player manipulative game which had all aspects of the baseball game. Student interest 4.

28. Engineer, Softside.
Thinking game which requires building a bridge in the fewest moves. Student interest 2.

29. Starwars - trench fight, Softside.
Challenging manipulative game which involves avoiding fighters with until the target appears. Student interest 3.

The following programs were investigated but not used, usually because of the following: length in loading, length of playing time, program investigated but not put on student tape in time for the sessions we have just finished, teacher oriented program, program too difficult, or program not useful at this time.

PROGRAM NAME AND SOURCE	FUTURE USE	REASONS NOT USED.
1. Micro chess, Personal software		time to load and play long.
2. Electric Paint Brush, personal software Grds.6,7,8,9		must learn difficult instructions
3. Invasion Force, Radio Shack		difficult instructions time.
4. Invasion Orion, Automated simulations		difficult, time
5. Warloads, Speak Easy Software		not on tape in time.
6. Life two, Softside company	✓	not on tape in time.
7. Towers of Hanoi, Softside	✓	not on tape in time. Good logic game
8. Livingston, Softside		time - adventure game
9. Golf, Softside		not interesting
10. Dead stick, softside		not on tape, manipulat
11. Haiku, 101 Basic computer games	✓	pre lesson needed.
12. Picture Draw, 80 U.S.	✓	pre lesson needed
13. Treasure Dungeon, Softside		time - adventure
14. Ghost Sub, 80 U.S.		pre lesson needed guess angles.
15. Yhatze, Creative Computing		game better played wit cards.
16. Organ, C load	✓	need music amplifier.
17. Checkers, Radio Shack		game better played oth way.
18. Othello, Local		game better played oth way
19. Santa Paravia and Fuimaccio, Instant software	✓	was played a few times
20. Adventure, Personal software		time, difficulty, adventure.
21. Tarzan, Local adaptation		needs censorship
22. Code Breaker, local	✓	similar to Mastermind
23. Romulan, Starwars, Starlanes		not on tape for student
24. Kriegspiel, Softside Company		time, wargame

PROGRAM NAME AND SOURCE	FUTURE USE	REASONS NOT USED AND T
25. Magician, C load	✔	not on tape, puzzle
26. Galactic trades,		difficult to win, time
27. Space War		not on student tape.
28. Space Maze, Creative computing		uses Level III, difficu
29. Around the Horn, Softside		time, boring
30. Golf, 101 Basic computer games		Boring
31. Midway, Softside		not on student tape, m ipulation
32, Battle ship, C load		better played another
33. Reaction time, measurement C load		not on student tape.
34. L game, C load	✔	not on student tape puzzle.
35. Shopping Spree C load	✔	not on student tape simulation
36. Cyborg Wars		time, wargame
37. Up Periscope		time, war game
38. Planet miners		time, simulation
39. Periodic tables, C load	7,8,9 Grades	pre lesson needed
40. Dancing Demon Radio Shack	✔	not on student tape good potential.
41. C.I.A. Adventure, Softside		time, adventure
42. TIAR, C load	Grades 7,8,9	puzzle.
43. Escape C load		not on student tape
44. Radio Shack - Basic Course PartI		time on computer.
45. Micro Movie, Radio Shack		time, pre lesson neede
46. Lost Dutchmans Mine, Softside		time, adventure
47. Mad Lib, local modification		loading time
48. Level III, Microsoft		extra-language parts.
49. Amazing, 101 computer games		not on student tape
50. Lander, Radio Shack manual		difficult simulation
51. Chase Five, softside		boring
52. Space Taxi, Radio Shack		not on student tape
53. Mastermind, a number of various		better played another
54. Hi Q (1) C load		better played another

PROGRAM NAME AND SOURCE	FUTURE USE	REASONS NOT USED & TYP
55. Chomps, C load	✓	similar to Nim
56. Morse Code, Softside		Pre lesson needed.
57. One aimed Bandit, C load		manipulative
58. Intersteller Navigation, Local	Jr. Sec.	shows star positions from other locations.
59. Personality profile	Jr. Sec.	not on student tape
60. Various games from 101 computer games. Most short, not interesting. Bug jump, oust, corral, awari, life, nim, tic tac toe.		
61. Various business tapes, Mortgage, etc.		not on student tape
62. Various Math functions	Jr. Sec.	not on student tape

APPENDIX E

PERIODICALS: We have read or skimmed the following periodicals for information or programs which we thought might be of value.

MAGAZINE AND REVIEW	ISSUES TO READ
1. Byte for advanced hobbyist, some articles on education very few useable software programs, but great for learning advanced concepts. All computers	No. 1 to present
2. Creative Computing for general hobbyist, some articles on education, some useable software, simulations, games, columns on each of the major computers dealing with reviews of software and short programming ideas. Good magazine for general public who wish to learn about computers.	Sept/77 to presen
3. 80 Micro Computing for general TRS 80 hobbiest. Some articles on education. Some useable software, also some hardware modifications for TRS 80. Best magazine for TRS 80 owner who wants to learn about his machine.	No1 to present
4. 80 U.S. for general TRS 80 hobbyist. Some good software. Some hardware modifications. Good magazine for learning about TRS 80	various issues borrowed from pre No. 80 subscription Nov.80
5. Kilobaud for general hobbyist. Some articles on education. Good for hardware. Some software is useful.	Most issues borro 1978 to present.

PERIODICALS (continued)

6. Interface Age for general hobbyist with business interest. Some articles on education (see Bob Albrects colmn on "My TRS 80 likes me...)**	various issues borrowed 1979 to present
7. Softside General TRS 80, Apple, Atari, Hobbyist. Deals mostly with software. Can be bought as pre-recorded programs but most of the soft side programs we have were labourously typed in.	various issues borrowed 1979 to present. Subscription Nov.80.
8. C load Magazine TRS 80 software magazine on tape programs on various topics. None we saw had the advanced graphics that Softside is known for.	Nov. 1979/80
9. The Computer Teacher Good for the teacher beginning with computers in education. Some software. Each issue seems to improve	80/81 Subscriptio: Nov. 80.
10. Radio Electronics	1977 to present
11. Popular Electronics Computer articles are becoming increasingly more common in Electronics magazines. Some of these are of interest to the general teacher hobbyist.	1977 to present
12. Other magazines: General;Personal computing, Recreational Computing, ROM, On Computing Dr. Dobbs,.., General Hobbyist, smaller circulation therefore fewer articles.	various issues bor rowed or bought usually to find ou how the magazine w set up.
13. Other Magazines: Apple	latest issue bough as school district is buying Apple co puters.

BOOKS:

We have read or skimmed the following books for information or programs which we thought might be of value.

1. Pathways Through The ROM; George Blank,
   Advanced hobbyist who needs to know the internal code of TRS80 Basic so that he can write assembly language routines.

2. Users Manual For Level I Radio Shack, 1977

3. Users Manual For Level II Radio Shack, 1978
   beginning hobbyist who wishes to learn the basic language as used by the TRS-80 Computer.

4. TRS 80 Micro Computer Technical Reference Handbook, Radio Shack, 1978
   for the advanced hobbyist who wishes to understand the TRS80 hardware so that he can make modifications or trouble shoot when something goes wrong with the computer.

5. TRS-80 Assembly Language Programming, W. B. Arden Jr., Radio Shack, 1979.
   for the person who wishes to learn about assembly language.

6. Advanced Basic, J. S. Coan, Hayden Book Company, Inc., 1977.

7. Basic Computer Games, edited D. H. Ahl, Creative Computing Press, 1978.
   a good book for software of a not too advanced nature. Games tend to not be very sophisticated.

8. Computers and Society - 2nd Edition. S. Rothman, C. Mosmann
   Science Research Associates, 1976
   Good overview of what computers mean to society and how they work but is now out of date because of the micro computer revolution.

9. More Basic Computer Games, D. H. Ahl, Creative Computing Press, 1979
   software in printed form.

10. Understanding Digital Computers, F. M. Mims, Radio Shack, 1978
    Good for understanding about machine language.

BIBLIOGRAPHY

ARTIST AND COMPUTER Edited by R. Peauitt, Creative Computing Press, 1976.
This book shows various pictures produced by artists using a computer. The text at a Sr. High or College Level.

THE COMPLETE COMPUTER Van Tussel, Science Research Associates, 1976.
This book provides interesting articles, poetry and fiction written about the computer. Might be good for the teacher making up a computer literacy program for Jr. and Sr. High.

COMPUTERS AND YOUNG CHILDREN W. A. R. Chambers, J. Murray, J. Wiley, Nuffield Foundation, 1972.
This b ok discusses flow charts that could be made by young children, as well as how to set up a class room so that it functions as a human computer using computer instruction such as STORE (1) MLT SUB. (The system is similar to Cardiac used by Bell Canada to introduce computer functions.)

DISCOVERING BASIC R. E. Smith, Hayden Book Company, Inc., 1970
This book has many short programs for learning basic but is more directed at J. Sr. or College because of its emphasis on math theory.

GAME PLAYING WITH BASIC, D. D. Spencer, Hayden Book Company, inc., 1977.
This book starts off with a brief review of Basic words then leads the reader through a number of simple games such as guess the number. There is also a section of problems that the reader can try to solve by writing Basic Programs.
This book might be good for Jr. or Sr. High.

INTRODUCTION TO COMPUTER PROGRAMMING IN THE LANGUAGE BASIC 5/6/7 STUDENT WORKBOOK. W. Dodds, Vancouver School Board, 1980.
This booklet gives a brief introduction to Basic as used on the Pet computer . A good part of it is Machine dependent but it provides ideas for possible directions to go.

MORE BASIC COMPUTER GAMES Edited by D. H. Ahl, Creative Computing Press 1979.
This book contains many simple games which could be typed into the computer by someone who already knows the rudiments of Basic but wishes to learn some of the finer points such as Lefts $, Mid $, etc. Then it is a matter of making the program run.

ONE COMPUTER 30 KIDS K. Richardson, M. Etienne, Meka Publishing Co., 1980.
This book which we received just a few weeks ago (May) seems to be exactly what we were looking for at the beginning of the year. It contains the Basic words that we taught our children and uses a good manner of question - answer. The book will still need teacher help and intervention but will provide a good start for a computer literacy program.

PERSONAL COMPUTERS WHAT THEY ARE AND HOW TO USE THEM B. G. Wels, Prentice-Hall Inc., 1978.
This book presents a general overview of personal computers for the person who wishes to know a little bit before buying one. It is an early book in the field but does contain some informatio that the teacher could use for background when he or she is making up a computer Literacy program.

PROGRAMMING M. BASIC WITH APPLICATIONS T. Logsdon, Anaheim Publishing Company, 1977.
This book was originally set out as a course for College but could be adapted to lower levels as it goes relatively slowly through the Basic language.

PROGRAMMING IN BASIC FOR BUSINESS, B. Bosworth, H. L. Nagel, Science Research Associates Inc., 1977
This book has lots of short programs to illustrat each word in Basic.

PROGRAMMING TECHNIQUES FOR LEVEL II BASIC, W. Borden, Jr., Radio Shack 1980.
This book requires the reader to already know Basic but can be used as a teacher reference for some of the further embelishments it contains such as string packing, etc.

57 PRACTICAL PROGRAMMS & GAMES IN BASIC K. Tracton, Radio Shack, 1978
This book has many short programs in Basic, but is directed at those who already know a little. A few programs might be useful.

80 PROGRAMS FOR THE TRS-80 Edited by J. Perry & C. Brown, 80 Micro-computing, 1979.
This book contains a numberof programs that might be useful to a beginning programmer at a Jr. or Sr. High level. There are some Educational programs in this book such as Matching States with Capitals - (this could easily be changed to Provinces.)

THE BASIC COOKBOOK K. Tracton, Radio Shack, 1978.
This book is a good dictionary of Basic words but is not Machine dependent. Several words are mentioned which do not occur in TRS-80 Level II Basic, such as Mat statements.

COMPUTERS

· Now that We Know how to Use It ·

At Riverbend
We Love Our Computer

The world of the microcomputer arrived in October 1979, at River Bend School, courtesy of the Parents' Committee. The school spent four months getting acquainted with its Radio Shack TRS-80 by using games and short programs. Reg joined the Alberni Computer Club and was able to obtain numerous TRS-80 games.

An enriched group of grade five and six math students, under the direction of Mr. Nordman, used the computer for studying problems and problem solving strategies.

A six-week, K - 7, science and social studies unit, dealing with Computer Literacy, was launched in the Spring of 1980. The students scanned newspapers and magazines to form a scrapbook of computer items as an aid to generate computer awareness. We put together a resource package for teachers. From these and other materials, teachers led discussions of:

History of Computers
Language of Computers
Uses/Abuses of Computers
Parts of a Computer
Are Computers a Good Idea?

Students were extremely interested, often carrying these discussions home and returning to school with parents' views. As an individual project, students were instructed in flow charting and then they produced numerous flow charts of everyday occurences,. e.g. Primary area might chart brushing your teeth.'; the intermediate area might chart a fairy tale or how to do a dance step. This task demanded and developed logical thinking, sequencing, and clarity of presentation.

Each of the teachers at River Bend (Barbara Kelly, Angela Snikkers. Margaret McDonough, Jackie Levans) led their multi-age groups into various topics depending on interest. The computer was used in each group to illustrate some applications. Reg led one group in an initial computer science class using a guided discovery approach. As children became more adept they taught others. The high levels of interest and enthusiasm generated a need for a more indepth project for 1981.

Simultaneously, John Parker at Calgary School was experimenting with his own TRS-80 in his grade four class. He started with a few of the most necessary words such as PRINT, INPUT, RUN, LIST, and ENTER. He worked for three weeks until most children in one group had worked through three different programs using about eleven basic words. The words were discussed before they were used. John developed programs as they were needed.

To avoid duplication of efforts, John Parker moved to River Bend for the 1980/81 year. He and Reg started to design an elementary computer unit which can be used by any elementary teacher, regardless of background and expertise.

PRESENT AND FUTURE

This year we expect to have an extensive computer program separated into three areas: from September to Christmas we will be working with the grade six and seven students; from January to Easter the grade four and five students; and from April to June the grade two and three students.

The first thing that we did to plan the year was search the periodicals: Interface Age, Kilobaud Microcomputing, '80 Microcomputing Byte, Creative Computing, and others for programs and ideas that would be appropriate for either teacher background or direct student use.

The best articles for the elementary school that we found were in Bob Albrecht's column in Interface Age called "My TRS-80 Likes Me When I Teach Kids How To Use It". It starts from the very beginning basic words and gets increasingly more involved in each colum.

As well as searching the literature, we mad up a sequential pack of computer job cards which we hope will act like a dictionary of basic words. We will be making up another group of job cards which the students will use as a basis for writing their own programs.

Since we are able to borrow two computers we are able to slot children into individual twenty-minute periods during the language arts and arithmetic periods. This will allow about fifty minutes a week or about six and one-half hours during the eight weeks per student.

Our program started this year on September 17th with twenty- five students. For the first two sessions we discussed the mathematical operations, the PRINT and RUN statement and how to use the shift and break keys. From then on the students worked at job cards. By September 22nd, two students understood what they were doing, sufficiently well to create their own simple programs by paraphrasing what they had learned. The next day there were six more students who had become programmers.

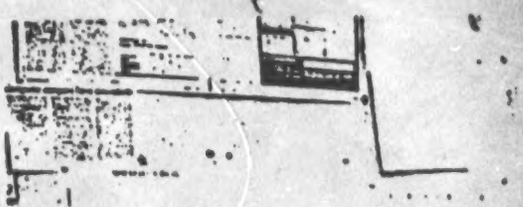
We observed that the students have shown a dramatic improvement in typing and program reading skills during the week.

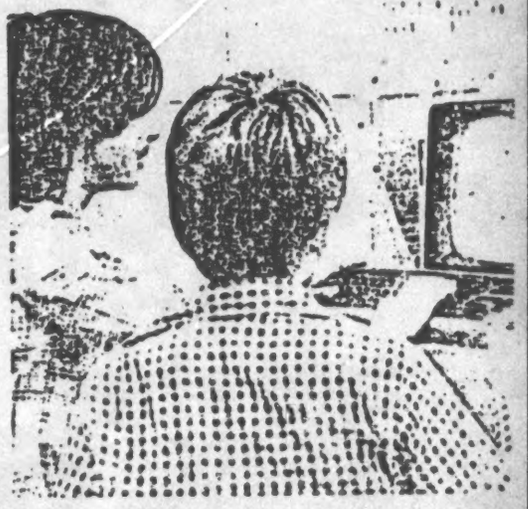
As of September 29th, five or six students were halfway through our proposed job cards and many others were not far behind. Students who are experiencing difficulty or who need extra instruction on drill in arithmetic skills are working on the third machine.

Not only are students enjoying working on the computer during arithmetic and language arts periods, but they are also coming in before school, at lunch time, and after school to type programs as well as play computer games.

It is hoped that by the end of the eight weeks some students will have written and saved programs which can be used with the lower grades. It is also expected that students will write programes to solve selected problems involving arithmetic, graphics, sound, and games.

Right now, students and teachers are having a fantastic teacher/student/machine relationship!





Type your name

Press ENTER.

Screen will show

SN ERROR - The machine doesn't understand you!

Type

PRINT "LUCY"
PRINT "CHARLIE"
PRINT "SNOOPY"
PRINT "LUKE SKYWALKER"

PRINT " YOUR NAME "

Now you will write a program.

Type

```
10 CLS
20 PRINT "LUCY";
30 GOTO 20
```

140

Type RUN

Had enough?

Press the "BREAK Key.

Substitute " your Name" for " Lucy"

(over)

" -.-.

The computer will write out

```
10 CLS
20 PRINT" YOUR NAME"
30 GOTO 20
```

LOOK OUT!

SN ERROR message means you need to press Q to get the machine under control.

USE BOXES! instead

"boxes" such as A$, B$, etc., which can hold names or other strings of characters.

Box!

```
10 CLS
20 LET A$ = "LUCY"
30 PRINT A$;
40 GOTO 30
RUN
```

Try other "boxes" such as B$, C$, N$, etc.

DON'T FORGET THE BREAK KEY. TO STOP THE ACTION

Use INPUT message

```
10 CLS
20 INPUT A$
30 PRINT A$;
40 GOTO 30
```

RUN

The computer should know A$ = "Lucy" or "HENRI". If not, how would you tell it?

Time Delays.

- Slow down the action. Use an "empty" FOR-NEXT loop as a time delay.

```
10 CLS
20 INPUT A$
30 PRINT A$;
40 FOR Z = 1 TO 100
50 NEXT Z
60 GOTO 30
```

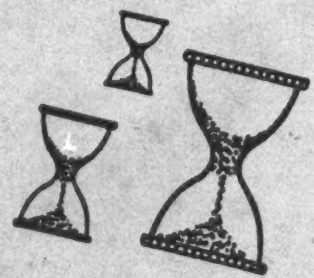
The FOR-NEXT loop (Lines 40 and 50) causes the TRS-80 to count from 1 to 100. Kids are usually surprised at how fast the computer can count to 100. Try other numbers (instead of 100) in the FOR statement.

Slower: Use numbers greater than 100.

Faster: Use numbers less than 100.

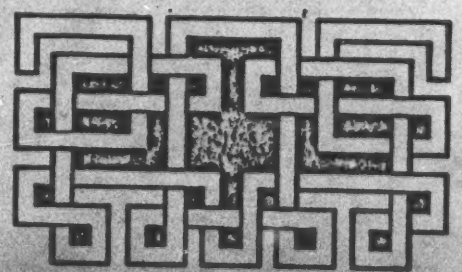
- Hmmm . . . how about a variable time delay?

```
10 CLS
20 INPUT A$
30 PRINT A$;
40 LET T = 100
50 FOR Z = 1 to T
60 NEXT Z
70 GOTO 30
```



Now change T by replacing line 40 with:

```
40 LET T = 200
```

Review card 3
The "BLINKING LIGHT"

4

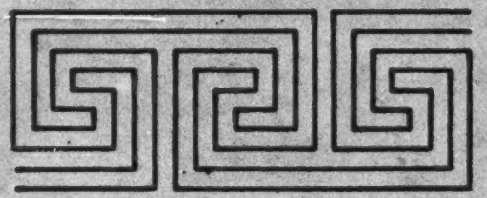
- We can blink a name on and off.

```
10 CLS
20 INPUT A$

30 CLS
40 PRINT A$;
50 LET T=100
60 FOR Z=1 TO T
70 NEXT Z

80 CLS
90 LET T=100
100 FOR Z=1 TO T
110 NEXT Z

120 GOTO 30
```

Lines 40–70: Name ON

Lines 80–110: Name OFF

To slow down the action, make T>100. To speed it up, make T<100.



SUBROUTINES
Review card 4

5

- Our program has two identical time delays. Great opportunity to introduce SUBROUTINES!

```
10 CLS
20 INPUT A$

40 CLS
50 PRINT A$;
60 GOSUB 100

70 CLS
80 GOSUB 100

90 GOTO 40

100 LET T=100
110 FOR Z=1 TO T
120 NEXT Z
130 RETURN
```

♦ Save room for Line 30 (below)

Lines 40–60: Name ON

Lines 70–80: Name OFF

PRINT @

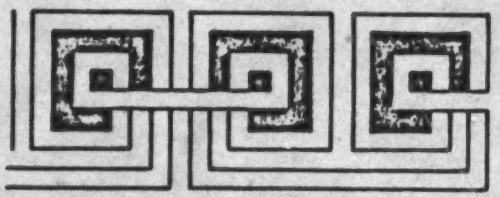
. In the above program, replace Line 50 with one of these. In fact, do all of them, one by one.

```
50 PRINT @0,A$;
50 PRINT @32,A$;
50 PRINT @64,A$;
50 PRINT @500,A$;
```

and so on.

Look up Screen map. There are 1024 print positions numbered 0 to 1023. This program puts your name where you want it on the screen. Notice Line 20 uses an A$ Box (See card 2)

```
10 CLS
20 INPUT "WHAT IS YOUR NAME";A$
30 INPUT "WHERE SHOULD I BLINK IT";N

40 CLS
50 PRINT @N,A$;          Name ON
60 GOSUB 100             at position N

70 CLS                   Name OFF
80 GOSUB 100             (everywhere)

90 GOTO 30

100 LET T=100
110 FOR Z=1 TO T
120 NEXT Z
130 RETURN
```

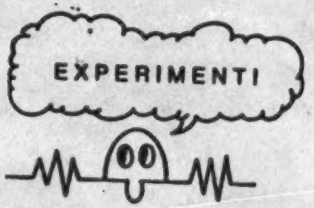



- As Tom Lehrer might say, "If you are getting bored by this stuff, you have only yourself to blame; you should never have let me begin." And so, I will reluctantly end with just one more program, which causes a name to scamper across the screen from upper left to lower right, then again and again and . . .

```
10 CLS
20 INPUT "WHAT IS YOUR NAME";A$

30 FOR K=0 TO 1023
40 CLS
50 PRINT @K,A$;
60 NEXT K

70 GOTO 30
```

Too fast? Probably—so, you put in a time delay to slow down the action.

See card 3

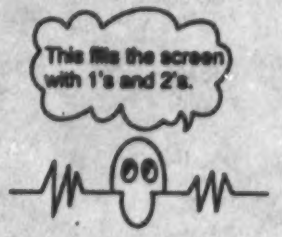


RND statement (Random no.) '8

Experiment!

Do some experiments.

```
10 CLS
20 PRINT RND(2);
30 GOTO 20
```




```
10 CLS
20 PRINT RND(3);
30 GOTO 20
```

```
10 CLS
20 PRINT RND(6);
30 GOTO 20
```

```
10 CLS
20 PRINT RND(10) - 1;
30 GOTO 20
```

Review card '8

OK, so RND gives random numbers. So what? Well, let's use them to put a name here, there and everywhere on the screen. Here for a moment, there for a moment, somewhere else for a moment.

```
10 CLS
20 X = RND(1023)
30 PRINT @X,"LUCY";
40 T = 100
50 FOR Z = 1 TO T          } Time delay
60 NEXT Z
70 GOTO 10
```

Explain lines 20 and 30 carefully. Line 20 gives a random integer from 1 to 1023, inclusive, and puts this number in box X. Line 30 causes the name "LUCY" to be printed at position X on the screen. Lines 50 and 60 are a time delay. Change Line 40 to decrease or increase the delay.

Faster: Make T<100

Slower: Make T>100

Let each kid change Line 30 to his or her name. Why let Lucy get all the glory? To make it easier, try the following program.

```
10 CLS
20 INPUT "WHAT IS YOUR NAME";N$
30 CLS
40 X = RND(1023)
50 PRINT @X,N$;
60 T = 100
70 FOR Z = 1 TO T
80 NEXT Z
90 GOTO 30
```

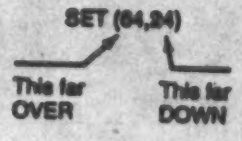
As usual, slowly and patiently explain what is happening, to your eager young learners. If they say, "What if . . . ?" then you say, "Try it and find out!"

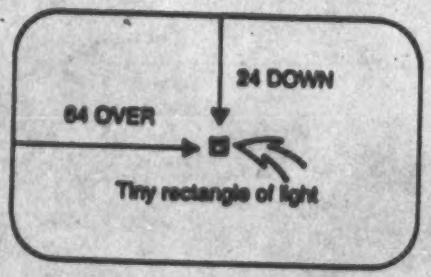
Patterns

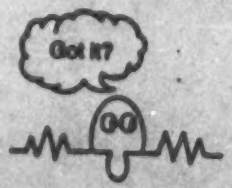
Patterns! . . . That's what we will do this time. Show you how to put patterns on the screen, using tiny rectangles of light. And, we will do some "computer art," using the familiar RND function and a new BASIC statement called SET.

- SET turns on a tiny rectangle of light somewhere on the screen.

Somewhere on the screen? Well, of course, we must tell the TRS-80 where to turn on the tiny rectangle of light.



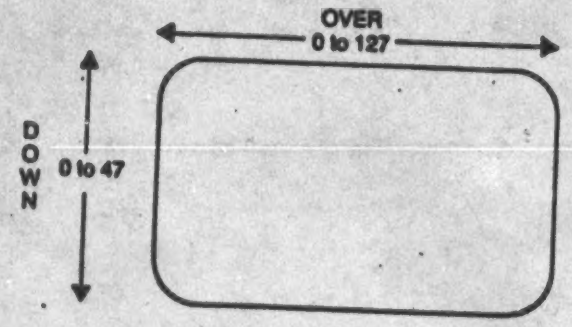




over

Think of it like this: SET (OVER, DOWN)

- OVER can be a whole number, 0 to 127.
- DOWN can be a whole number, 0 to 47.



Experiment!

Try these to get the hang of things.

- Press CLEAR, then press ENTER
- Type SET (0,0) and press ENTER
  Where is the tiny rectangle?
- Type SET (127,0) and press ENTER
  Where is the tiny rectangle?
- Type SET (0,47) and press ENTER
  Where is the tiny rectangle?
- Type SET (127,47) and press ENTER
  Where is the tiny rectangle?

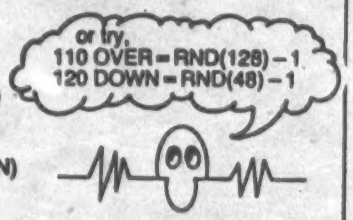
Now, for each of the following, guess where the rectangle will appear before you press ENTER. Then do it . . . press ENTER . . . were you close?

- Type SET (0,24) and press ENTER
- Type SET (127,24) and press ENTER
- Type SET (64,0) and press ENTER
- Type SET (64,47) and press ENTER
- Type SET (64,24) and press ENTER
- Type SET (32,12) and press ENTER

Starfall

Now, pretend that the screen is the sky. Also, pretend that each tiny rectangle of light is a star appearing in the night sky. Make it happen by storing and running this program.

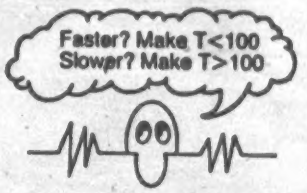
```
NEW
100 CLS
110 OVER = RND(127)
120 DOWN = RND(47)
130 SET (OVER,DOWN)
140 GOTO 110
```



When you RUN this program, the sky will begin filling with stars. Try to find . . . constellations, shapes, patterns . . . as stars turn on.

Too fast? Add a time delay.

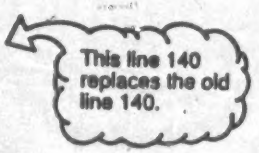
```
133 T = 100
135 FOR Z = 1 TO T
137 NEXT Z
```



Or, here is a trick you can use to stop the action momentarily while you stargaze, then continue letting more stars come out.

Add these lines to the program.

```
140 IF INKEY$ = "" THEN 110
150 IF INKEY$ = "" THEN 150
160 GOTO 110
```

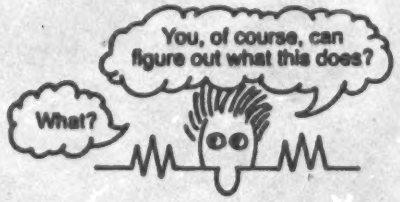


Have you added the above lines (140, 150, 160) to the original program? Yes . . . good! Now, RUN the modified program. To stop it, press any key; then, to continue, press any key. Now you can stop time! Or restart it! Happy stargazing.

Walkabout

Suppose you start near the center of the screen (OVER = 64, DOWN = 24) and walk (stagger?) at random. Up? Down? Left? Right? Try this program.

```
NEW
100 CLS
110 OVER = 64          Start near the center
120 DOWN = 24          of the screen
130 SET (OVER,DOWN)    Turn on a light
140 A = RND(3) - 2
150 B = RND(3) - 2
```




```
160 OVER = OVER + A    Well, obviously (?), this
170 DOWN = DOWN + B    will change OVER and
                       DOWN.
180 GOTO 130
```

Dismay not! Try this program. It will start near the center of the screen and . . . meander. Eventually, it will (probably) stop with an error message.

?FC ERROR IN 130

That happens because we tried to wander off the screen. Off the screen? Well, that means: OVER became less than 0 or more than 127 or DOWN became less than 0 or more than 47. More about that in a minute . . .

(cont.)

In the meantime, try this program.

```
NEW
100 CLS
110 X = RND(128) - 1   X will mean 'over.'
120 Y = RND(48) - 1    Y will mean 'down.'
130 SET (X,Y)
140 SET (X,47 - Y)
150 SET (127 - X,Y)    Remember: the first
160 SET (127 - X,47 - Y)  number means OVER;
                       the second number
170 GOTO 110           means DOWN.
```

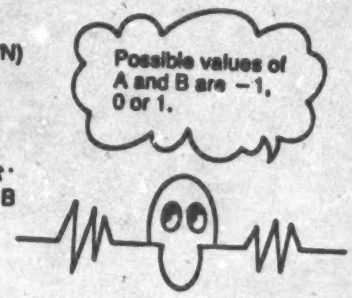
Run it for awhile, then press BREAK, RUN it again, etc. . . .

magic!

More Walkabout

Here again is our "walkabout" program.

```
100 CLS

110 OVER=64
120 DOWN=24

130 SET (OVER,DOWN)

140 A=RND(3)-2
150 B=RND(3)-2

160 OVER=OVER+A
170 DOWN=DOWN+B

180 GOTO 130
```



Remember, it eventually tries to wander off the screen. This happens if OVER<0 or OVER>127 or DOWN<0 or DOWN>47. When this happens, the TRS-80 stops with the following error message:


```
?FC ERROR IN 130
```

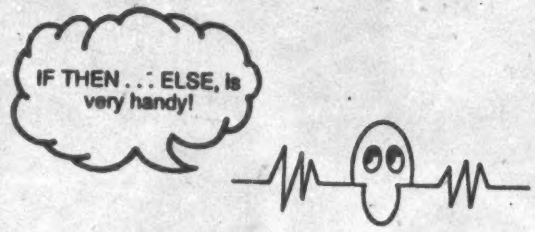
Aha! Let's fix that little bug in our program. Here is our bug tranquilizer. Add it to the program.

```
163 IF OVER<0 THEN OVER=OVER+1
167 IF OVER>127 THEN OVER =OVER-1
173 IF DOWN>0 THEN DOWN=DOWN+1
177 IF DOWN>47 THEN DOWN=DOWN-1
```

These additions to the program will sometimes seem to make the screen image "bounce along" against the edge of the screen. Try it. As the pattern develops, what shapes do you see? A castle? A dragon? A flower? An angular cloud? Buildings on a hillside?

Wouldn't it be nice to be able to stop the computer and look at the pattern on the screen? Or photograph it? Easy to do. Just add the following statements.

```
180 IF INKEY$=" "THEN 130

190 IF INKEY$=" "THEN 190 ELSE 130
```



Now, having added the above lines 180 and 190, RUN the program. To stop it, press any key (except BREAK!). Gaze upon the pattern on the screen. Then, press any key (except BREAK) and the pattern will continue growing.

INTERLUDE—A GAME

Once upon a time, computers were veerryyyy expensive. Many of them "talked" to people by means of a rather large, and definitely noisy, device called a . . . (suspense) . . . Teletype®!

The Teletype (or TTY, as it became affectionately known) typed information on paper. From these almost prehistoric times, we bring you a simple computer game enjoyed by multitudes of children—a game called NUMBER or, if you prefer, GUESS MY NUMBER. Our version of NUMBER, shown below, is modified slightly (but only slightly) for TRS-80.

```
100 REM***NUMBER, A NUMBER GUESSING GAME
    FOR TRS-80

200 REM***TELL PLAYER ABOUT THE GAME
210 CLS
220 PRINT "I AM THINKING OF A NUMBER FROM
    1 TO 100."
230 PRINT "GUESS MY NUMBER!!!"

300 REM***COMPUTER 'THINKS' OF A NUMBER, X
310 RANDOM
320 X = RND(100)

400 REM***GET GUESS, G, COMPARE WITH X
410 INPUT "YOUR GUESS"; G
420 IF G = X THEN 610
430 IF G<X THEN PRINT "TRY BIGGER"
440 IF G>X THEN PRINT "TRY SMALLER"
450 GOTO 410

600 REM***WINNER!
610 PRINT "THAT'S IT!!! MY NUMBER IS" X
620 FOR T = 1 TO 2000:NEXT
630 GOTO 210
```

Well, try the above program. It uses the screen abysmally, doesn't it? This program is fine for printed output, but terrible for TV. The printer is a one-dimensional medium; the TV screen is a two-dimensional medium.

over

So, here is a new version of NUMBER which illustrates a different way to use the TV screen.

```
100 REM***NUMBER, A NUMBER GUESSING GAME FOR TRS-80
110 REM***COPYRIGHT 1979 BY BOB ALBRECHT

200 REM***TELL PLAYER ABOUT THE GAME
210 CLS
220 PRINT @268, "I AM THINKING OF A NUMBER FROM 1 TO 100."
230 PRINT @396, "GUESS MY NUMBER! WHEN I ASK FOR YOUR"
240 PRINT @524, "GUESS, TYPE A NUMBER AND THEN PRESS THE"
250 PRINT @652, "BIG 'ENTER' KEY."
260 PRINT @780, "PRESS THE 'ENTER' KEY AND I WILL BEGIN."
270 IF INKEY$ = "" THEN 270 ELSE 310

300 REM***COMPUTE SECRET, A RANDOM NUMBER FROM 1 TO 100
310 RANDOM
320 SECRET = RND(100)

400 REM***GET GUESS, COMPARE WITH SECRET (UP TO 13 GUESSES)
410 CLS
420 FOR N = 0 TO 12
430   PRINT @ 64*N,"" ;
440   INPUT "YOUR GUESS"; GUESS
450   IF GUESS = SECRET THEN 610
460   IF GUESS<SECRET THEN PRINT @ 64*N + 20, "TRY BIGGER"
470   IF GUESS>SECRET THEN PRINT @ 64*N + 20, "TRY SMALLER"
480 NEXT N

500 REM***OOPS! RAN OUT OF GUESSES — LIMIT IS 13
510 PRINT "OOPS! THAT'S 13 GUESSES, MY NUMBER IS" SECRET
520 PRINT "TO PLAY AGAIN, PRESS THE 'ENTER' KEY."
530 IF INKEY$ = "" THEN 530 ELSE 210

600 REM ***WINNER! WATCH THE STARS COME OUT!
610 CLS
620 FOR STARS = 1 TO 100
630   PRINT @RND(1023),"*";
640 NEXT STARS
650 PRINT @406, "MY NUMBER IS" SECRET
660 PRINT @529, "YOU GOT IT IN" N + 1 "GUESSES!!!"
670 PRINT @656, "TO PLAY AGAIN: PRESS 'ENTER' "
680 IF INKEY$ = "" THEN 680 ELSE 210

999 END
```

NAME BLINKER

```
10 CLS
20 INPUT "WHAT IS YOUR NAME";A$
30 INPUT "WHERE SHOULD I BLINK IT";N
40 CLS
50 PRINT@N,A$;          Name ON
60 GOSUB 100            at position N
70 CLS                  Name OFF
80 GOSUB 100            (everywhere)
90 GOTO 40              Do it again
100 LET T = 100         Time delay
110 FOR Z = 1 TO T
120 NEXT Z
130 RETURN
```

When the computer asks WHERE SHALL I BLINK IT? You enter a number from 0 to 1023. What happens if you don't enter an integer from 0 to 1023? Try it and find out. Especially if you are working with kids, have them try it and find out. Always: Experiment!

NAMERUNNER

```
10 CLS
20 INPUT "WHAT IS YOUR NAME";A$
30 FORK = 0 TO 1023
40 CLS
50 PRINT @K,A$;
60 NEXT K
70 GOTO 30
```

160

Try it. Too fast? Probably, so add a time delay, as follows.

```
55 FOR Z = 1 TO 10 : NEXT Z
```

Slower yet? Try

```
55 FOR Z = 1 TO 99 : NEXT Z
```


HYPERSPACE NAME

RUN this program. Your name will appear somewhere on the screen, then disappear (into Hyperspace?), reappear, disappear, reappear. . .here, there, anywhere on the screen.

```
10 CLS
20 INPUT "WHAT IS YOUR NAME";N$
30 CLS
40 X = RND(1023)
50 PRINT@X,N$;
60 T = 100
70 FOR Z = 1 TO T
80 NEXT Z
90 GOTO 30
```

Hmmm. . .wonder what would happen if we change line 90 to 90 GOTO 40. Do it!

STARFALL

```
100 CLS
110 OVER = RND(127)
120 DOWN = RND(47)
130 SET(OVER,DOWN)
140 IF INKEY$=""THEN 110
150 IF INKEY$=""THEN 150
160 GOTO 110
```

In lines 140 and 150, do not put a space between the quotation marks.

```
140 IF INKEY$ =""THEN 110
```


When there is nothing between the quotes ("") this is an empty string. It is much like the empty set. There is nothing between quotation marks — the empty string is a nothing string. The empty set contains no elements.

RUN STARFALL. The screen will begin filling with "stars." To stop the action, press almost any key. Stargaze, if you wish. Then, press almost any key. More stars will appear.

161

WALKABOUT

```
100 CLS

110 OVER = 64
120 DOWN = 24

130 SET(OVER,DOWN)

140 A = RND(3) - 2
150 B = RND(3) - 2

160 OVER = OVER + A
163 IF OVER < 0 THEN OVER = OVER + 1
167 IF OVER > 127 THEN OVER = OVER - 1

170 DOWN = DOWN + B
173 IF DOWN < 0 THEN DOWN = DOWN + 1
177 IF DOWN > 47 THEN DOWN = DOWN - 1

180 IF INKEY$ = "" THEN 130
190 IF INKEY$ = "" THEN 190 ELSE 130
```

Try this program. Watch what happens. Perhaps it is a meandering painter, going sometimes left, sometimes right, sometimes up, sometimes down.

What do you see? An angular cloud? A castle? A crystal flower? A city on a hill? If you see something you like, press almost any key. The computer will stop. . .the pattern remains on the screen. Look at it, photograph it, whatever. Then press almost any key. The pattern continues to grow.

HAPPY BIRTHDAY, MOTHER

```
100 REM**A BIRTHDAY CARD FOR MOTHER
110 CLS

200 REM**DRAW LINES TOP AND BOTTOM
210 FOR OVER = 0 TO 127
220 SET(OVER, 0)
230 SET(OVER, 47)
240 NEXT OVER

300 REM**DRAW LINES ON LEFT AND RIGHT
310 FOR DOWN = 0 TO 47
320 SET(0, DOWN)
330 SET(127, DOWN)
340 NEXT DOWN

400REM**THE HAPPY BIRTHDAY MESSAGE
410 PRINT @ 348, "HAPPY";
420 PRINT @ 475, "BIRTHDAY";
430 PRINT @ 604, "MOTHER";

500 REM**WAIT A BIT, THEN DO IT AGAIN
510 T = 1000
520 FOR Z = 1 TO T : NEXT Z
530 GOTO 110
```

You, of course, can change the message to MERRY CHRISTMAS, SANTA or HAPPY UNBIRTHDAY, GANDALF or ???

Or change lines 210 and 310 as follows.

```
210 FOR OVER = 0 TO 127 STEP 2
310 FOR DOWN = 0 TO 47 STEP 2
```


SIMPLE SCRIBBLE

```
100 REM**SCRIBBLE ON THE SCREEN
110 CLS
120 OVER = 64
130 DOWN = 24

200 REM**TURN ON A LIGHT AT OVER, DOWN
210 SET(OVER,DOWN)

300 REM**WAIT FOR SOMEONE TO PRESS A KEY
310 K$ = INKEY$ : IF K$ = "" THEN 310

400 REM**IF KEY WAS R OR L, CHANGE OVER
410 IF K$ = "R" THEN OVER = OVER + 1
420 IF K$ = "L" THEN OVER = OVER - 1

500 REM**IF KEY WAS U OR D, CHANGE DOWN
510 IF K$ = "D" THEN DOWN = DOWN + 1
520 IF K$ = "U" THEN DOWN = DOWN - 1

600 GOTO 210
```

When you RUN this program, a single tiny light will come on near the center of the screen. Press the 'R' key a few times. The TRS-80 moves the light to the right, and paints a line along the way.

Then press 'D' a few times. The line prints downward. OK, try 'L' for left and 'U' for up. Got it? Good. . .scribble away.

Beware! If you go too far right, or too far left, or too far up or too far down, you will get an ? FC ERROR IN LINE 210. We will fix that someday, sometime.

```
100 REM** WANDERING STAR
110 CLS

200 REM**COSMIC DUST
210 FOR K = 1 TO 200
220 PRINT @ RND(1022),".";
230 NEXT K

300 REM**WANDERING STAR APPEARS
310 X=480
320 PRINT @X,"*";

400 REM**WANDERING STAR RESTS
410 T = 2000
420 FOR Z=1 TO T : NEXT Z

500 REM**WANDERING STAR WANDERS
510 R=RND(4)
520 IF R=1 THEN X2=X+1
530 IF R=2 THEN X2=X-1
540 IF R=3 THEN X2=X+64
550 IF R=4 THEN X2=X-64
560 PRINT @X," ";
570 IF X2<0 OR X2>1022 THEN 910
580 X=X2
590 PRINT @X,"*";
600 T = 200
610 FOR Z = 1 TO T : NEXT Z
620 GOTO 510

900 REM**WANDERING STAR DISAPPEARS
910 PRINT @0, "OOPS! WANDERING STAR HAS"
920 PRINT "LEFT THIS UNIVERSE."
930 GOTO 930
```


Eventually, wandering star reaches the edge of the screen, and disappears. Despair not! Wandering Star will return.

```
110 CLS
200 REM***COSMIC DUST
210 FOR K = 1 TO 200
220 PRINT @ RND(1022), ".";
230 NEXT K

300 REM***WANDERING STAR APPEARS
310 ROW = 7
320 COL = 32
330 PRINT @(64*ROW + COL), "*";

400 REM***WANDERING STAR RESTS
410 T = 2000
420 FOR Z = 1 TO T : NEXT Z

500 REM***WANDERING STAR WANDERS
510 PRINT @(64*ROW + COL), " " ;     She leaves her old
                                     place
520 W = RND(4)
530 IF W = 1 THEN ROW = ROW + 1   She might go down
540 IF W = 2 THEN ROW = ROW - 1   She might go up
550 IF W = 3 THEN COL = COL + 1   She might go right
560 IF W = 4 THEN COL = COL - 1   She might go left

600 REM***DID SHE GO OFF-SCREEN?
610 IF ROW < 0 OR ROW > 15 THEN 910
620 IF COL < 0 OR COL > 63 THEN 910
630 IF 64*ROW + COL = 1023 THEN 910

700 REM***NO, SHE IS STILL ON-SCREEN
710 PRINT @(64*ROW + COL), "*" ;
720 T = 100
730 FOR Z = 1 TO T : NEXT Z
740 GOTO 510
```



```
900 REM***WANDERING STAR GOES OFF-SCREEN
910 PRINT @ ), "WANDERING STAR HAS LEFT THIS
                UNIVERSE."
920 PRINT "FAREWELL, WANDERING STAR."
930 GOTO 930
```